

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA:  
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:  
Ingenieros de Sistemas**

**TEMA:  
CREACIÓN DE UNA SOLUCIÓN INFORMÁTICA QUE PERMITA LA  
VISUALIZACIÓN DE DATOS RELACIONADO CON EL EXCESO DE  
VELOCIDAD DEL TRANSPORTE INTERPROVINCIAL EN LA RUTA  
QUITO – RIOBAMBA, UTILIZANDO GEOLOCALIZACIÓN EN UNA  
APLICACIÓN MÓVIL PARA GENERAR DATOS ABIERTOS**

**AUTORES:  
HENRY GONZALO GÓMEZ NARANJO  
DANIEL ALEJANDRO YANGUA CORREA**

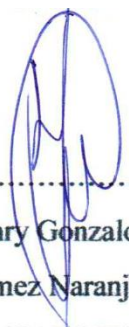
**TUTOR:  
FRANKLIN EDMUNDO HURTADO LARREA**

**Quito, febrero del 2019**


## **CESION DE DERECHOS DE AUTOR**

Nosotros, HENRY GONZALO GÓMEZ NARANJO con documento de identificación N° 171610750-1, y DANIEL ALEJANDRO YANGUA CORREA con el documento de identificación N° 172359561-5, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: “CREACIÓN DE UNA SOLUCIÓN INFORMÁTICA QUE PERMITA LA VISUALIZACIÓN DE DATOS RELACIONADO CON EL EXCESO DE VELOCIDAD DEL TRANSPORTE INTERPROVINCIAL EN LA RUTA QUITO – RIOBAMBA, UTILIZANDO GEOLOCALIZACIÓN EN UNA APLICACIÓN MÓVIL PARA GENERAR DATOS ABIERTOS”, mismo que ha sido desarrollado para optar por el título de INGENIEROS DE SISTEMAS en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En la aplicación a lo determinado en la Ley de propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



Henry Gonzalo  
Gómez Naranjo  
CI: 1716107501



Daniel Alejandro  
Yangua Correa  
CI: 1723595615

Quito, febrero del 2019

## **DECLARATORIA DE COAUTORIA DEL TUTOR**

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Proyecto Técnico, con el tema: “CREACIÓN DE UNA SOLUCIÓN INFORMÁTICA QUE PERMITA LA VISUALIZACIÓN DE DATOS RELACIONADO CON EL EXCESO DE VELOCIDAD DEL TRANSPORTE INTERPROVINCIAL EN LA RUTA QUITO – RIOBAMBA, UTILIZANDO GEOLOCALIZACIÓN EN UNA APLICACIÓN MÓVIL PARA GENERAR DATOS ABIERTOS”, realizado por HENRY GONZALO GÓMEZ NARANJO y DANIEL ALEJANDRO YANGUA, obtenido un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, febrero del 2019



.....

FRANKLIN EDMUNDO HURTADO LARREA  
CI: 1713382016

## **DEDICATORIA**

Se lo decido a mis padres, Gonzalo y María Luisa, por haberme apoyado incondicionalmente en uno de mis mayores retos en la vida, a mis hermanos Jhonathan y Diana, que sin ellos no hubiese podido lograr este objetivo, gracias a ellos porque han sido como unos padres para mí, y nunca podre describir con palabras el agradecimiento y el cariño que tengo hacia ellos. Por último, a las personas que siempre me han apoyado, esta es la demostración que el trabajo y el sacrificio dan sus frutos.

Henry Gonzalo Gómez Naranjo

A todos los estudiantes de ingeniera amates al código, a mis padres, hermanos, novia y amigos quienes han estado apoyándome incondicionalmente, permitiéndome mantenerme siempre motivado para lograr este objetivo, por último, al Ing. José Luis Tarapues por haberme mostrado lo hermoso que puede ser la informática.

Daniel Alejandro Yangua Correa

## **AGRADECIMIENTO**

Agradecemos a la Universidad Politécnica Salesiana, la cual nos ha ayudado a formarnos no solo a nivel académico, sino como personas.

Al ingeniero Franklin Hurtado Larrea, el cual ha sido una fuente de inspiración, gracias por haber sido una parte importante dentro de nuestra formación, llevando a cabo una catedra excelente en todos los años que fue nuestro docente, y por ayudarnos y guiarnos en el desarrollo del trabajo de titulación.

Henry Gonzalo Gómez Naranjo

Daniel Alejandro Yangua Correa

## ÍNDICE

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>Objetivo General .....</b>	<b>2</b>
<b>Objetivos Específicos .....</b>	<b>2</b>
<b>Capítulo 1 Marco Teórico .....</b>	<b>7</b>
1.1 Ingeniería de tránsito.....	7
1.2 Metodología de desarrollo de software .....	7
1.3 Arquitectura de software.....	8
1.4 UML.....	9
1.5 Tecnologías y lenguajes de programación .....	9
1.5.1 PHP (Hypertext Preprocessor) .....	9
1.5.2 Javascript.....	9
1.5.3 Java .....	10
1.5.4 HTML .....	10
1.5.5 CSS .....	10
1.5.6 Node js .....	11
1.5.7 Express.js .....	11
1.5.8 CodeIgniter.....	11
1.5.9 MySQL.....	12
1.5.10 Android .....	12
1.5.11 Leaflet .....	12
1.5.12 Retrofit .....	12
1.6 Herramientas de desarrollo de software.....	13
1.6.1 Visual Studio Code .....	13
1.6.2 Android Studio .....	13
1.6.3 JMeter.....	13
1.7 Estructura de datos .....	13
1.7.1 Tipos de Datos Estructurados.....	14
1.8 Estilo Arquitectónico de la Aplicación .....	15
1.8.1 Interfaz de programación de aplicaciones (API).....	15
1.8.2 REST (Representational State Transfer) .....	15
1.8.3 MVC .....	16
1.8.4 MVP .....	17
1.9 Ciencia de Datos .....	17
1.10 Open Data .....	17
1.11 Pruebas .....	17
1.11.1 Pruebas unitarias .....	17
1.11.2 Pruebas de rendimiento .....	18
1.11.3 Pruebas de estrés. ....	18
1.11.4 Pruebas de funcionalidad .....	18
<b>Capítulo 2 Análisis y diseño .....</b>	<b>19</b>
2.1 Requerimientos del sistema.....	19
2.1.1 Requerimientos funcionales de las aplicaciones móvil y web. ....	19
2.1.2 Requerimientos no funcionales de la aplicación web. ....	20
2.1.3 Requerimientos arquitectónicos de las aplicaciones móvil y web. ....	20
2.1.4 Requerimientos de rendimiento de las aplicaciones móvil y web. ....	20
2.2 Diagrama de casos de uso. ....	20
2.2.1 Caso de uso perfil usuario aplicación móvil. ....	21
2.2.2 Caso de uso perfil usuario aplicación web.....	22
2.3 Análisis de arquitectura.....	22

2.3.1 Arquitectura inicial. ....	23
2.3.2 Diagrama de arquitectura planteado. ....	24
2.4 Análisis de proceso planteado. ....	25
2.5 Diagrama de clases. ....	26
2.5.1 Diagrama de clases aplicación móvil. ....	26
2.5.2 Diagrama de clases servicios web. ....	27
2.5.3 Diagrama de clases aplicación web. ....	28
2.6 Diagrama de base de datos. ....	29
2.7 Diseño de interfaces. ....	31
2.7.1 Diseño interfaz web. ....	31
2.7.1 Diseño interfaz móvil. ....	32
<b>Capítulo 3 Construcción y pruebas</b> .....	<b>33</b>
3.1 Arquitectura del sistema. ....	33
3.2 Arquitectura de la aplicación web. ....	34
3.2.1 Estándares Aplicación Web .....	35
3.2.2 Componentes del sistema web .....	35
3.2.3 Servicios Web .....	35
3.3 Arquitectura de la aplicación móvil .....	36
3.3.1 Estándares aplicación móvil. ....	36
3.3.2 Componentes de la aplicación móvil .....	36
3.3.3 Permisos en AndroidManifest.xml. ....	37
3.3.4 Declaración para la ejecución de servicios en AndroidManifest.xml .....	38
3.4 Código Relevante del Sistema .....	38
3.4.1 Verificación de red WiFi para el envío de datos desde el dispositivo móvil .....	38
3.4.2 Uso del GPS del equipo móvil .....	39
3.4.3 Implementación de un servicio con GPS .....	41
3.4.4 Cálculo de la velocidad entre dos distancias con el GPS del móvil. ....	42
3.4.5 Envío de datos de forma asíncrona desde el móvil .....	43
3.4.6 Cálculo de la media de las velocidades aplicativo Web .....	44
3.5 Pruebas .....	45
3.5.1 Implementación de las aplicaciones para el escenario de pruebas. ....	45
3.5.2 Plan de pruebas .....	46
3.5.3 Pruebas Unitarias .....	46
3.5.3.1 Obtener la lista de unidades de transporte. ....	47
3.5.3.2 Verificar la existencia de un usuario .....	48
3.5.3.3 Verificación de la creación de punto de georreferenciado. ....	49
3.5.3.4 Ejemplo de test esperando un error .....	50
3.5.3 Pruebas de Rendimiento. ....	51
3.5.4 Pruebas de Estrés. ....	55
3.5.5 Pruebas de Funcionalidad .....	56
3.5.5.1 Pruebas de funcionalidad en la aplicación web. ....	56
3.5.5.2 Pruebas de funcionalidad de la aplicación móvil. ....	63
3.5.6 Pruebas de usabilidad. ....	68
<b>CONCLUSIONES</b> .....	<b>74</b>
<b>RECOMENDACIONES</b> .....	<b>75</b>
<b>LISTA DE REFERENCIAS</b> .....	<b>76</b>
<b>ANEXOS</b> .....	<b>80</b>

## INDICE DE FIGURAS

Figura 1 Diagrama del patrón MVC .....	16
Figura 2 Caso de uso perfil usuario aplicación móvil.....	21
Figura 3 Caso de uso perfil usuario aplicación web .....	22
Figura 4 Diagrama inicial de componentes.....	23
Figura 5 Diagrama final de componentes .....	24
Figura 6 Diagrama proceso planteado.....	25
Figura 7 Diagrama de clases aplicación móvil .....	26
Figura 8 Diagrama de clases servicios web .....	27
Figura 9 Diagrama de clases aplicación web .....	28
Figura 10 Diagrama lógico de base de datos versión final .....	30
Figura 11 Diseño interfaz web .....	31
Figura 12 Diseño interfaz móvil .....	32
Figura 13 Diagrama de arquitectura del sistema.....	33
Figura 14 Diagrama Modelo Vista Controlador .....	34
Figura 15 Ejemplo de servicio web para obtener las unidades de transporte .....	36
Figura 16 Permisos de aplicación en AndroidManifest.xml .....	37
Figura 17 Declaración de servicios en AndroidManifest.xml .....	38
Figura 18 Método de verificación de red WiFi para el envío de datos.....	39
Figura 19 Implementación de la clase LocationListener .....	40
Figura 20 Implementación de un servicio con GPS .....	41
Figura 21 Líneas de código del cálculo de la velocidad .....	42
Figura 22 Envío de datos de manera asíncrona al servidor.....	43
Figura 23 Función del cálculo medio de las velocidades.....	44
Figura 24 Porción de código para cargar el mapa con los datos del viaje correspondiente .	45
Figura 25 Test unitarios lista de unidades de transporte .....	47
Figura 26 Resultado obtenido del test.....	47
Figura 27 Verificación de la disponibilidad de nombre de usuario (alias) .....	48
Figura 28 Resultado obtenido del test de verificación de usuarios.....	48
Figura 29 Verificación de la creación de punto de georreferenciado .....	49
Figura 30 Resultado obtenido verificación de la creación de punto de georreferenciado ...	50
Figura 31 Ejemplo de test esperando un error .....	50
Figura 32 Resultado de la pregunta 1, encuesta aplicación web.....	69
Figura 33 Resultado de la pregunta 2, encuesta aplicación web.....	69
Figura 34 Resultado de la pregunta 3, encuesta aplicación web.....	70
Figura 35 Resultado de la pregunta 4, encuesta aplicación web.....	70
Figura 36 Resultado de la pregunta 5, encuesta aplicación web.....	71
Figura 37 Resultado de la pregunta 1, encuesta aplicación móvil. ....	71
Figura 38 Resultado de la pregunta 2, encuesta aplicación móvil .....	72
Figura 39 Resultado de la pregunta 3, encuesta aplicación móvil .....	72
Figura 40 Resultado de la pregunta 4, encuesta aplicación móvil .....	73



## INDICE DE TABLAS

Tabla 1 Tablas de base de datos, versión final.....	29
Tabla 2 Prueba de rendimiento con carga de un usuario (primera parte) .....	51
Tabla 3 Prueba de rendimiento con carga de un usuario (segunda parte).....	51
Tabla 4 Prueba de rendimiento con carga de 50 usuarios (primera parte).....	52
Tabla 5 Prueba de rendimiento con carga de 50 usuarios (segunda parte) .....	52
Tabla 6 Prueba de rendimiento con carga de 100 usuarios (primera parte).....	53
Tabla 7 Prueba de rendimiento con carga de 100 usuarios (segunda parte) .....	53
Tabla 8 Prueba de rendimiento con carga de 250 usuarios (primera parte).....	54
Tabla 9 Prueba de rendimiento con carga de 250 usuarios (segunda parte) .....	54
Tabla 10 Datos de escenarios para el análisis de las pruebas de estrés.....	55
Tabla 11 Pruebas funcionales de la aplicación web .....	57
Tabla 12 Pruebas funcionales de la aplicación móvil .....	63

## **RESUMEN**

El desarrollo del proyecto recopila el conocimiento recibido a través de la carrera de Ingeniería de Sistemas, el documento detalla el proceso del desarrollo de software de una solución informática que consta de una aplicación móvil, la cual es la encargada de capturar los datos del exceso de velocidad de los buses interprovinciales de la ruta Quito – Riobamba, y enviarlos a través de servicios web, para almacenarlos en una base de datos, y posteriormente ser consultados por una aplicación web. La carretera Quito-Riobamba fue seleccionada para llevar a cabo el sistema ya que presenta condiciones en las que los transportes interprovinciales tienden a concurrir en el exceso de velocidad como infracción de otras normas de tránsito.

## **ABSTRACT**

The development of the project collects the knowledge acquired through the career of Systems Engineering. The following document details the process of software development of a computer solution consisting of a mobile application, which is responsible for capturing the data of excess speed of the interprovincial buses on the route Quito - Riobamba, thus sends the data through web services to be store in a database, and later be consulted by a web application. The Quito-Riobamba highway was selected to carry out the system as it presents conditions in which interprovincial transports tend to have an excess speed as a violation of other traffic regulations

## INTRODUCCIÓN

Ecuador ha aumentado la tasa de fallecidos respecto al número de accidentes de tránsito en el primer semestre del año 2018, respecto al mismo periodo en los últimos cuatro años, según estadísticas de la Agencia Nacional de Tránsito, estableciendo una cifra de 8.49 fallecidos por cada 100 accidentes ocurridos en el territorio nacional. (universo, 2018)

La ruta Quito – Riobamba aparenta características que se proyecta para que los conductores cometan infracciones, entre las principales son conducir el vehículo superando los límites máximos de velocidad, no respetar las señales de tránsito, conducir desatentos a las condiciones de tránsito siendo esta la principal causa de accidentes a nivel nacional. Los datos del mes de febrero del año 2018 muestran que en la provincia de Pichincha han ocurrido 629 accidentes, mientras que en la provincia de Chimborazo han ocurrido 72 accidentes. (ANT, 2018)

Actualmente existen numerosas aplicaciones móviles con las que se puede obtener la velocidad a la cual el dispositivo se está moviendo, sirven para monitorizar cualquier tipo de transporte público como privado, aun así, no existe una integración con una aplicación web en la cual se pueda observar cual ha sido la velocidad en todo el trayecto, y que además sea de acceso gratuito y libre para cualquier usuario.

Existe un control de velocidad con GPS que se está utilizando en la ciudad de Guayaquil, en el cual una vez que se haya excedido la velocidad máxima permitida, la unidad de transporte será automáticamente multada, este sistema se está utilizando en el transporte urbano de la ciudad, aunque no se puede visualizar la velocidad a la cual circula el vehículo. (Comercio, 2018)

## **Objetivo General**

Crear una solución informática que permita la visualización de datos relacionados con el exceso de velocidad del transporte interprovincial en la ruta Quito - Riobamba, utilizando geolocalización en una aplicación móvil y la utilización de una base de datos para el almacenamiento de la información, la cual se mostrará en una aplicación web.

## **Objetivos Específicos**

Diseñar una estructura de datos que permita almacenar y consultar la información de forma efectiva.

Construir una aplicación web que permita la recepción de datos, almacenamiento y consulta de la base de datos.

Construir una aplicación móvil sobre Android con lenguaje Java que permita registrar de manera automática el exceso de velocidad del autobús.

Llevar a cabo pruebas de integración de la aplicación tanto de rendimiento, seguridad y en usabilidad.

## **Justificación**

Para coadyuvar en la erradicación de esta problemática se propone un control de velocidad con la participación de los ciudadanos quienes serán los que actúen como sensores y colaboren registrar los datos relativos al exceso de velocidad en las unidades de transporte público interprovincial en la que ellos se encuentren, los datos se pretende que sean para uso público tanto para su estudio como para la toma de decisiones y la generación de política pública, de esta manera intentar mitigar con un problema que afecta a todo el país y que anualmente genera decenas de muertos en las carreteras interprovinciales y las calles de las ciudades.

Según el diario El Comercio, el Gobierno trabaja constantemente en mejorar la forma de controlar las vías, para el año 2018 se colocará 500 radares más, aparte de los ya existentes, en las carreteras de todo el país, llevando a cabo una inversión de 50 millones de dólares, para tratar de mitigar el número de infracciones de tránsito. La solución propuesta en este proyecto permitirá que el usuario sea el principal actor recolectando y enviando datos, de esta manera ser un instrumento más para la detección. (comercio, 2018)

Para ello se creó una aplicación móvil en la que los usuarios puedan recolectar información acerca con la posición en la cual se encuentran, y la velocidad a la cual se están desplazando. Esta información es presentada en una aplicación web en la cual se puede realizar consultas sobre la información obtenida. La solución se llevó a cabo usando tecnologías existentes como Android, PHP, JavaScript, MYSQL, entre las principales.

### **Propuesta de solución y alcance**

La propuesta planteada fue la creación de una solución informática, que engloba diferentes componentes para la visualización de datos referentes a la velocidad del transporte interprovincial en la ruta Quito - Riobamba, para ello se deben tener en cuenta los componentes que son necesarios para el desarrollo del proyecto.

En primer lugar, una aplicación móvil, con la que se obtendrá la velocidad en el transporte público, para ello se utilizará la tecnología GPS (sistema de posicionamiento global) que ayudará a obtener la velocidad a la cual se desplaza el usuario en ese momento sobre el autobús, utilizando dos puntos geolocalizados, y llevando a cabo un cálculo entre la distancia de estos dos puntos, y el tiempo que ha tardado en registrar dichos puntos, de esta manera se obtiene la velocidad de desplazamiento en ese instante. Esta información será capturada por las personas que viajen en ese momento

en el medio de transporte interprovincial, y una vez capturado será enviado a un servidor el cual organizará y procesará los datos.

Cuando se tenga la información en el servidor, se utiliza una base de datos, la cual almacena los datos obtenidos de los diferentes equipos móviles que se encuentren en el medio de transporte, y de esta manera tener dicha información para ser mostrada mediante una aplicación web.

El segundo componente de la solución sería la aplicación web implementada sobre un servidor de pruebas, la cual en un futuro pueda estar disponible para la ciudadanía, empresas, o entidades públicas, que así lo deseen.

En este componente se conocerá los resultados obtenidos de la recolección de datos, así como del procesamiento de dicha información, todo ello a través de los parámetros de la cooperativa de transporte, la unidad, la fecha, y el viaje del vehículo, y así ofrecer los resultados en un mapa interactivo, en el cual se visualizará en que puntos de latitud y longitud se ha superado el límite de velocidad máximo establecido por la ley orgánica de transporte terrestre, tránsito y seguridad vial.

Cabe destacar que la manipulación de las aplicaciones, así como de la base de datos, puede afectar a los resultados finales, es por ello por lo que el uso de los códigos fuentes, únicamente se pueden manipular por personal autorizado, y con conocimientos avanzados en programación, sino fuese así, no se garantiza que las aplicaciones funcionen correctamente.

## **Marco metodológico**

### **Scrum**

Para el desarrollo del sistema se ha optado por la metodología Scrum, esta metodología ágil permite crear pequeños incrementos de software que son el resultado de los llamados sprints (micro proyectos), esta es la razón principal para optar por el uso de esta metodología. La aplicación de la metodología Scrum permite iterar y adaptar los próximos sprints a los resultados que se obtienen al finalizar cada uno de ellos, esto provee un ritmo de trabajo sostenible.

Otro de los motivos por el que se opta por la metodología Scrum, es el hecho que se tiene una idea de cómo son los requerimientos del proyecto, y su alcance, pero debido a la naturaleza del software no se conocía con precisión el detalle de cada uno de los requerimientos, es por ello que se llevó a cabo un modelo incremental iterativo, con el que se puede escalar y adaptarse a las necesidades de cada requerimiento.

Para el uso de esta metodología se han utilizado los siguientes roles:

Equipo Scrum: El equipo Scrum ha sido conformado por los integrantes del proyecto de titulación.

Scrum master: para el papel de Scrum master se ha designado a cada uno de los integrantes del equipo, ya que de esta manera se permite a cada uno de los integrantes liderar y planificar las actividades del proyecto, dando diferentes puntos de vista para el desarrollo.



## **Sprints.**

Para llevar a cabo el desarrollo del proyecto, se establecieron el uso de 3 horas diarias de lunes a viernes, y 8 horas los fines de semana para cumplir los siguientes sprints:

Sprint 1: Módulo de captura de datos a través de la aplicación móvil.

Sprint 2: Módulo de almacenamiento local en cada dispositivo móvil, y envío de datos al servidor.

Sprint 3: Módulo de servicios web para recepción de datos de aplicación móvil.

Sprint 4: Módulo de selección de datos para generar reportes en aplicación web.

Sprint 5: Módulo de presentación de datos en un mapa georreferenciado.

Sprint 6: Pruebas finales.

## **Reuniones.**

En cada reunión se conocía los avances de los sprints llevados a cabo por el equipo Scrum, así mismo se conocía como estaba iterando el proyecto, ya que, al no existir un precedente de un proyecto similar, se debía concretar de la manera más precisa posible los requerimientos del siguiente sprint.

Las reuniones se realizaron cada 15 días, estableciendo así un periodo de dos semanas para poder finalizar cada uno de los sprints,

## **Capítulo 1**

### **Marco Teórico**

#### **1.1 Ingeniería de tránsito**

La Ingeniería de Tránsito contempla conocimientos útiles en la solución de problemas de circulación en vías en entorno urbano y rural.

La Ingeniería de Tránsito, se centra en aspectos importantes como el volumen y la velocidad que se interrelacionan en la teoría de flujo de tránsito y continúa con una aplicación particular a la formación de colas, posteriormente con herramientas y procedimientos para valorar los parámetros del tránsito a través de los estudios técnicos de tránsito que producen los insumos para el análisis operacional, su aplicación en el servicio y capacidad en algunos tipos de infraestructuras como las carreteras de dos carriles que representan la mayoría de las carreteras rurales y los semáforos como los dispositivos de control del tránsito más comunes, se sigue con el tratamiento de diseños complementarios a las vías como su señalización y estacionamientos. Tratando un problema gravísimo en la circulación como es la accidentalidad. (Alaix, 2007)

#### **1.2 Metodología de desarrollo de software**

La metodología es una colección de buenas prácticas para incrementar las posibilidades del éxito dentro de un proyecto de software, la metodología ofrece herramientas de trabajo para estructurar, planificar y controlar el proceso del desarrollo del software.

Es por ello que el uso de una metodología para el desarrollo de software es totalmente fundamental, y es por ello que se ha utilizado una metodología incremental iterativa, la cual nos ayuda a crecer de manera controlada, estableciendo ciertas iteraciones que

deben cumplirse en un determinado tiempo, y que no se puede avanzar hasta la siguiente iteración, hasta que no se haya cumplido la iteración correspondiente, de esta manera se logra mitigar el encontrarse con problemas que hayan surgido al inicio del proyecto, cuando este ya se haya culminado.

Otro de los puntos importantes es la escalabilidad dentro del proyecto, ya que así, los componentes pueden evolucionar, ya que se pueden agregar más opciones de requerimientos una vez que ya se haya empezado el proyecto.

“Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo” (Tinoco Gómez, O., Rosales López, P. P., & Salas Bacalla, J., 2010)

### **1.3 Arquitectura de software**

Existen diferentes definiciones para la Arquitectura de Software, llegando a existir hasta tres tipos de definiciones distintas, planteadas por diferentes autores, una de las más aceptadas dentro del mundo del desarrollo de sistemas sería la forma en que diferentes componentes se disponen para llevar a cabo una tarea con éxito.

“La AS (Arquitectura de Software) es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o

diferimiento del detalle inherente a la mayor parte de las abstracciones.” (Reynoso, 2004)

## **1.4 UML**

El Lenguaje Unificado de Modelamiento es un lenguaje de modelado de sistemas de software, el cual sirve para especificar, documentar, y modelar un sistema.

“El modelado es esencial en la construcción de software para...

- Comunicar la estructura de un sistema complejo
- Especificar el comportamiento deseado del sistema
- Comprender de mejor manera el proceso de construcción de software.
- Descubrir oportunidades de simplificación y reutilización

Un modelo proporciona los planos de un sistema y puede ser más o menos detallado, en función de los elementos que sean relevantes en cada momento” (Booch, G., Rumbaugh, J., Jacobson, I., Martínez, J. S., & Molina, J. J. G., 1999)

## **1.5 Tecnologías y lenguajes de programación**

### **1.5.1 PHP (Hypertext Preprocessor)**

“Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML”. (secure.php.net, 2018)

PHP ha ido evolucionando de ser un lenguaje no orientado a objetos a incorporarlos en el lenguaje permitiendo mejor modularidad y aplicación de patrones de diseño, una de las principales ventajas es la facilidad con la que permite trabajar con etiquetas HTML además de estar presente en casi todos los servidores web.

### **1.5.2 Javascript**

“Es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado

en muchos entornos sin navegador, tales como node.js, Apache CouchDB y Adobe Acrobat. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de programación funcional, orientada a objetos e imperativa. “ (Mozilla.org, 2018)

Javascript al ser un lenguaje de script no necesita ser compilado al contrario de Java, hace uso de un intérprete para poder traducir las órdenes al motor en el que se ejecuta.

### **1.5.3 Java**

Java es un lenguaje de programación y una plataforma informática difundido por primera vez en 1995 por Sun Microsystems. Java es rápido, seguro y fiable, su uso se extiende desde aplicaciones móviles, escritorio, web hasta su incorporación en video juegos y súper computadoras. (Oracle, 2018)

### **1.5.4 HTML**

“Lenguaje de Marcado para Hipertextos (HyperText Markup Language) es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad.” (Mozilla.org, 2017)

### **1.5.5 CSS**

Hojas de estilo en cascada (Cascading Stylesheets) se usa para darle estilo y posicionarlo visualmente las etiquetas de una estructura HTML (HyperText Markup Language). CSS (Cascading Stylesheets) se puede usar, por ejemplo, para cambiar la fuente, el color, el tamaño y el espaciado del contenido, para formar múltiples columnas, añadir animaciones y otros elementos decorativos. (Mozilla.org, 2017)

### **1.5.6 Node js**

Concebido como un entorno de ejecución de JavaScript orientado a eventos asíncronos, Node está diseñado para construir aplicaciones escalables en red, se pueden manejar muchas conexiones concurrentes. Cuenta con un solo canal para los procesos de peticiones el cual funciona de manera no bloqueante esto permite la asincronía y evita que los procesos se manejen de manera síncrona. (Nodejs.org, 2018)

### **1.5.7 Express.js**

Una infraestructura web rápida, minimalista y flexible basado sobre el patrón MVC (Modelo-Vista-Controlador) para creación las aplicaciones web en Node.js. Por lo general, se prefiere “Express” a “Express.js”, aunque esta también se acepta. (Nodejs.org, 2018)

### **1.5.8 CodeIgniter**

CodeIgniter es un framework de PHP con un tamaño muy reducido en comparación a otros frameworks PHP como Laravel o Symfony, diseñado para el desarrollo de aplicaciones web, basadas en patrón arquitectónico MVC (Modelo-Vista-Controlador) que incorpora un conjunto de herramientas para mejorar la experiencia de desarrollo. Propone convenciones para la estructuración de proyectos, clases, controladores entre los principales. Una fuerte ventaja de CodeIgniter es que se puede alojar una aplicación desarrollada con este framework en un servidor que cuente con el lenguaje de programación PHP al contrario de otros frameworks que necesitan una configuración de librerías para que puedan ser alojados. (EllisLab, 2018)

### **1.5.9 MySQL**

“Una base de datos es una colección estructurada de datos. Puede ser cualquier cosa, desde una simple lista de compras hasta una galería de imágenes o la gran cantidad de información en una red corporativa. Para agregar, acceder y procesar datos almacenados en una base de datos informática, se necesita un sistema de administración de bases de datos como el servidor MySQL. Dado que las computadoras pueden para manejar grandes cantidades de datos se presenta como una solución los sistemas de administración de bases de datos que juegan un papel central en la informática, como utilidades independientes o como parte de otras aplicaciones. “ (MySQL , 2018)

### **1.5.10 Android**

Es un sistema operativo que mediante una interfaz de programación desarrollada en Java permite acceder a los componentes del sistema para la creación de aplicaciones y juegos innovadores de dispositivos móviles, se puede desarrollar en lenguajes de programación como Kotlin o Java. (Google, 2018)

### **1.5.11 Leaflet**

Biblioteca de JavaScript de código abierto para mapas interactivos aptos para dispositivos móviles, la cual tiene la mayoría de componentes que necesitan desarrolladores. Esta librería está diseñada tomando en cuenta la simplicidad, rendimiento y la facilidad de uso. (leaflet, 2018)

### **1.5.12 Retrofit**

Librería para el desarrollo de Android y Java, la cual es un cliente REST, que permite hacer peticiones GET, POST, PUT, PATCH, DELETE, HEAD, y descargar datos en formatos JSON o XML, de manera rápida y sencilla. (BEEVA, 2016)

## **1.6 Herramientas de desarrollo de software**

### **1.6.1 Visual Studio Code**

Visual Studio Code es un editor de código fuente ligero, una aplicación de escritorio multiplataforma para Windows, MacOS y Linux. Permite instalar extensiones para trabajar en otros lenguajes (como C ++, C #, Java, Python, PHP, Go) y tiempos de ejecución (como .NET y Unity). (Microsoft, 2018)

### **1.6.2 Android Studio**

Android Studio es un IDE (Entorno de Desarrollo Integrado) para el desarrollo de aplicaciones en Android y se basa en IntelliJ IDEA. Además del editor de códigos que colabora con la productividad y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan la productividad durante la compilación de aplicaciones móviles para Android. (Google, 2018)

### **1.6.3 JMeter**

Es un software de código abierto, una aplicación Java 100% pura diseñada para monitorear el comportamiento funcional de pruebas y medir el rendimiento. Originalmente fue diseñado para probar aplicaciones web, pero desde entonces se ha expandido para realizar pruebas de otras funciones. (Apache.org, 2018)

## **1.7 Estructura de datos**

Es una colección de datos que se caracteriza por su organización y las operaciones que se definen para ellos. Una estructura de datos vendrá caracterizada por las relaciones que surgen entre los datos y las posibles operaciones que son soportadas. (Quetglás, Lobo, & Cerverón Lleó, 1995)



## **1.7.1 Tipos de Datos Estructurados**

### **1.7.1.1 Cadenas**

Se define como una secuencia de caracteres que son interpretadas como un dato único, está formada por caracteres alfanuméricos y soporta diferentes operaciones según el lenguaje como la obtención de su longitud y comparación de cadenas entre las principales. (Quetglás, Lobo, & Cerverón Lleó, 1995)

### **1.7.1.2 Arrays**

Conjunto de datos del mismo tipo, almacenados en memoria del ordenador en posiciones adyacentes. Sus componentes individuales son llamados elementos y se distingue entre ellos por el nombre del arreglo seguido de uno o varios índices.

- Unidimensionales (vectores)
- Bidimensionales (matrices)

(Quetglás, Lobo, & Cerverón Lleó, Fundamentos de informática y programación, 1995)

### **1.7.1.3 Listas Enlazadas**

Está formada por un conjunto de nodos en lo que cada elemento contiene un puntero con la posición o dirección del siguiente elemento de la lista, los elementos de la lista están enlazadas por medio de los campos de enlaces. Esta característica obliga a que las listas sean almacenadas de forma adyacente para permitir que estén relacionadas entre sí. (Quetglás, Lobo, & Cerverón Lleó, Fundamentos de informática y programación, 1995)

### **1.7.1.4 Pilas**

Es una estructura lineal cuyos datos solo se pueden acceder por un extremo, denominado tope o cima, esta estructura tiene la peculiaridad de permitir solo 2

operaciones como introducir (push) o sacar (pop), los datos se almacenan en memoria y contienen un puntero denominado puntero de pilas que indica la posición del tope. (Quetglás, Lobo, & Cerverón Lleó, Fundamentos de informática y programación, 1995)

## **1.8 Estilo Arquitectónico de la Aplicación**

### **1.8.1 Interfaz de programación de aplicaciones (API)**

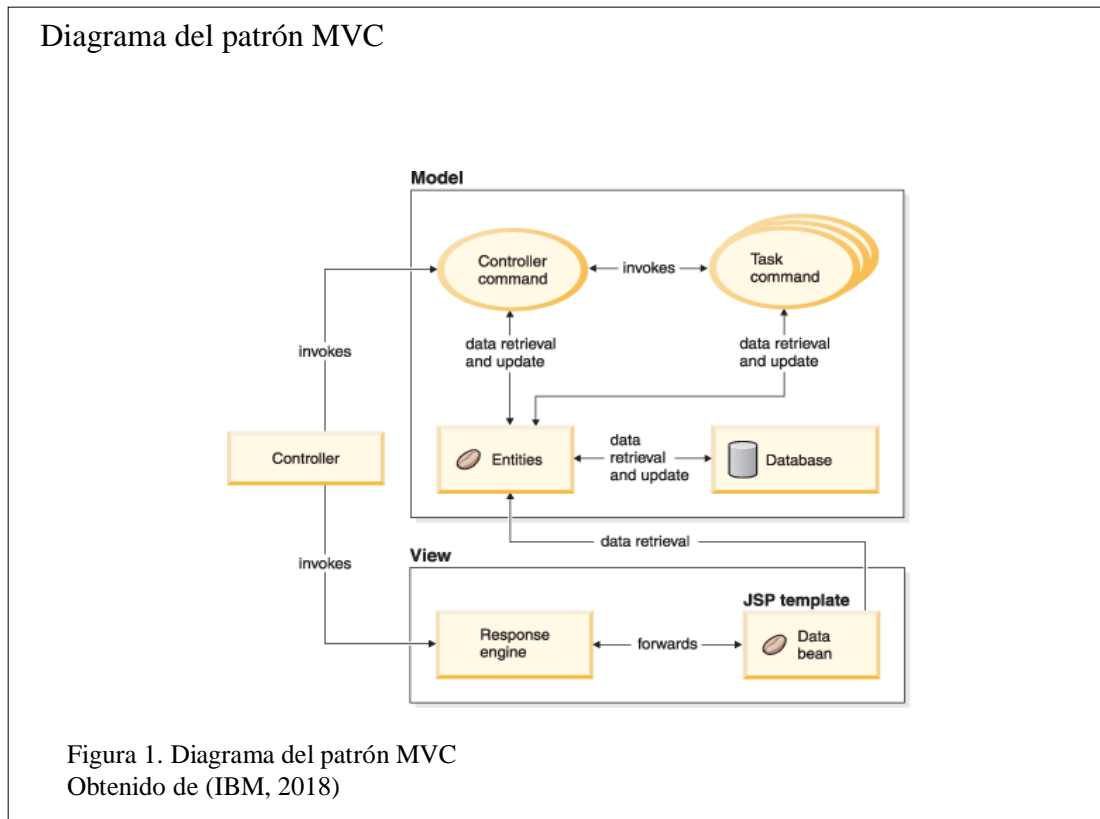
“Permite la relación entre dos aplicaciones para el intercambio de mensajes o datos, es un conjunto de funciones y procedimientos que ofrece una biblioteca para que otro software la utilice como capa de abstracción, un espacio de acceso e intercambio de información adicional en la parte superior”. (BBVA, 2016)

### **1.8.2 REST (Representational State Transfer)**

Se refiere a un grupo de restricciones de diseño de arquitectura de software que generan sistemas distribuidos eficientes, confiables y escalables. Un sistema se llama RESTful cuando se adhiere a esas restricciones. La idea básica de REST es que un recurso, por ejemplo, un documento, se transfiere con su estado y sus relaciones (hipertexto) a través de operaciones y formatos bien definidos y estandarizados. A menudo, las API o los servicios se llaman a sí mismos RESTful cuando modifican directamente un tipo de documento en lugar de desencadenar acciones en otro lugar. (Mozilla.org, 18)

### 1.8.3 MVC

El patrón de diseño de modelo-vista-controlador (MVC) especifica que una aplicación consta de un modelo de datos, de información de presentación y de información de control.



Como se aprecia el patrón en la figura No. 1, requiere que cada uno de estos elementos esté separado en distintos objetos. El modelo contiene únicamente los datos de la lógica del negocio que describe cómo pueden presentarse los datos a un usuario. La Vista presenta al usuario los datos del Modelo, no permite acceder a través de la Vista directamente sobre el Modelo para ello se hace uso del Controlador funcionando de capa intermedia. El Controlador escucha los sucesos desencadenados por la Vista (u otro origen externo) y ejecuta la reacción apropiada a estos sucesos. En la mayoría de los casos, la reacción es llamar a un método del Modelo. Puesto que la Vista y el

Modelo están conectados a través de un mecanismo de notificación, el resultado de esta acción se reflejará automáticamente en la Vista. (IBM, 2018)

#### **1.8.4 MVP**

La idea central de MVP (Modelo-Vista-Presentador) se basa en separar la Vista y la Lógica en un Presentador, el acoplamiento de la Vista y Modelo es realizada mediante el Presentador trabajando como intermediario entre estas dos capas. Las respuestas del presentador surgen en las acciones del usuario captadas por la Vista, son transferidas hacia el modelo y finalmente devuelven la información requerida hacia la Vista.(GU & TANG , 2010)

### **1.9 Ciencia de Datos**

La ciencia de datos es el proceso de descubrir información oculta en grandes cantidades de datos estructurados y no estructurados, utilizando métodos como las estadísticas, la minería de datos y la analítica predictiva. Esta área multidisciplinaria está cambiando el modo en que las organizaciones resuelven los problemas y ganan ventaja competitiva. (IBM, 2018)

#### **1.10 Open Data**

Se basa simplemente en dejar los datos generados abiertos al público sin ningún objetivo financiero permitiendo que se puede realizar análisis académicos, mejorar procesos, fomentar colaboración entre disciplinas, mantener disciplina e integridad fomentando la investigación en la ciudadanía. (Mozilla.org, 2018).

#### **1.11 Pruebas**

##### **1.11.1 Pruebas unitarias**

En las pruebas unitarias se lleva a cabo la comprobación de los componentes individuales, ya sean clases, funciones, módulos, etc. Cada componente se prueba de

manera individual, y así encontrar errores de código fuente, modelamiento de datos, o de diseño. (Testea, s.f.)

#### **1.11.2 Pruebas de rendimiento**

Las pruebas de rendimiento se utilizan para comprobar el funcionamiento del software en el momento de ejecución, estas pruebas se llevan a cabo durante todos los pasos del proceso de pruebas, inclusive en las pruebas unitarias, sin embargo, no es hasta que el sistema esté totalmente completo cuando se puede determinar el verdadero rendimiento de un sistema. (Pressman, R. S., & Troya, J. M., 1988)

#### **1.11.3 Pruebas de estrés.**

Las pruebas de esfuerzo, se realizan para enfrentar al sistema con situaciones anormales, en estas pruebas lo que se consigue es conocer hasta qué punto el software puede soportar una demanda excesiva de recursos, ya sea en cantidad, frecuencia, o volúmenes anormales. (Pressman, R. S., & Troya, J. M., 1988)

#### **1.11.4 Pruebas de funcionalidad**

Se basan usualmente en la verificación y validación de un conjunto de tareas que garantizan que el software implementa correctamente una función específica, este conjunto de diferentes tareas se llevan a cabo bajo los lineamientos deseados por el cliente y se determina si cumplió con los requerimientos esperados. (Pressman, R. S., & Troya, J. M., 1988)

## **Capítulo 2**

### **Análisis y diseño**

#### **2.1 Requerimientos del sistema**

El primer paso para llevar a cabo el proyecto fue el levantamiento de requerimientos, desde este punto se llevó a cabo un análisis de la solución planteada, en el cual se incluían los requerimientos funcionales y no funcionales de la solución, además se estableció la importancia de cada componente, y como deberían interactuar entre sí.

Al llevar a cabo el análisis de la solución, se encontró que la arquitectura de dicho sistema era de una gran importancia, por la forma en la cual se iban a disponer los componentes y como deben interactuar entre sí, es por ello que establecer la arquitectura fue un desafío al cual se tuvo que dar solución.

A continuación, se describe los requerimientos iniciales del proyecto.

##### **2.1.1 Requerimientos funcionales de las aplicaciones móvil y web.**

- Se almacenará la información obtenida en la aplicación móvil de forma local, hasta que el dispositivo móvil esté anclado a una red con conexión WIFI, de esta manera no se llevará a cabo el uso de datos móviles por parte del usuario móvil.
- Almacenar únicamente los datos relevantes para el estudio, de esta manera se discriminarán los datos que sean menores a la velocidad máxima genérica permitida por la Agencia Nacional de Tránsito, en la carretera Quito – Riobamba
- El acceso a la aplicación web no se debe llevar a cabo a través de un acceso con credenciales, ya que esta información es de acceso libre sin ningún tipo de restricción.

### **2.1.2 Requerimientos no funcionales de la aplicación web.**

- Utilización de mapa para la consulta de información por parte de la aplicación web, siendo una manera mucho más amigable de cara al usuario para consultar dicha información.

### **2.1.3 Requerimientos arquitectónicos de las aplicaciones móvil y web.**

- Aplicación móvil desarrollada en Android, la cual capturará los datos de velocidad, latitud, longitud, fecha y hora, del usuario de la aplicación móvil.
- Servidor, en el cual se almacenará y se procesará la información obtenida por el usuario móvil.
- Aplicación web, que se encargará de consultar la información almacenada en el servidor, y la mostrará amigablemente al usuario de la aplicación web.

### **2.1.4 Requerimientos de rendimiento de las aplicaciones móvil y web.**

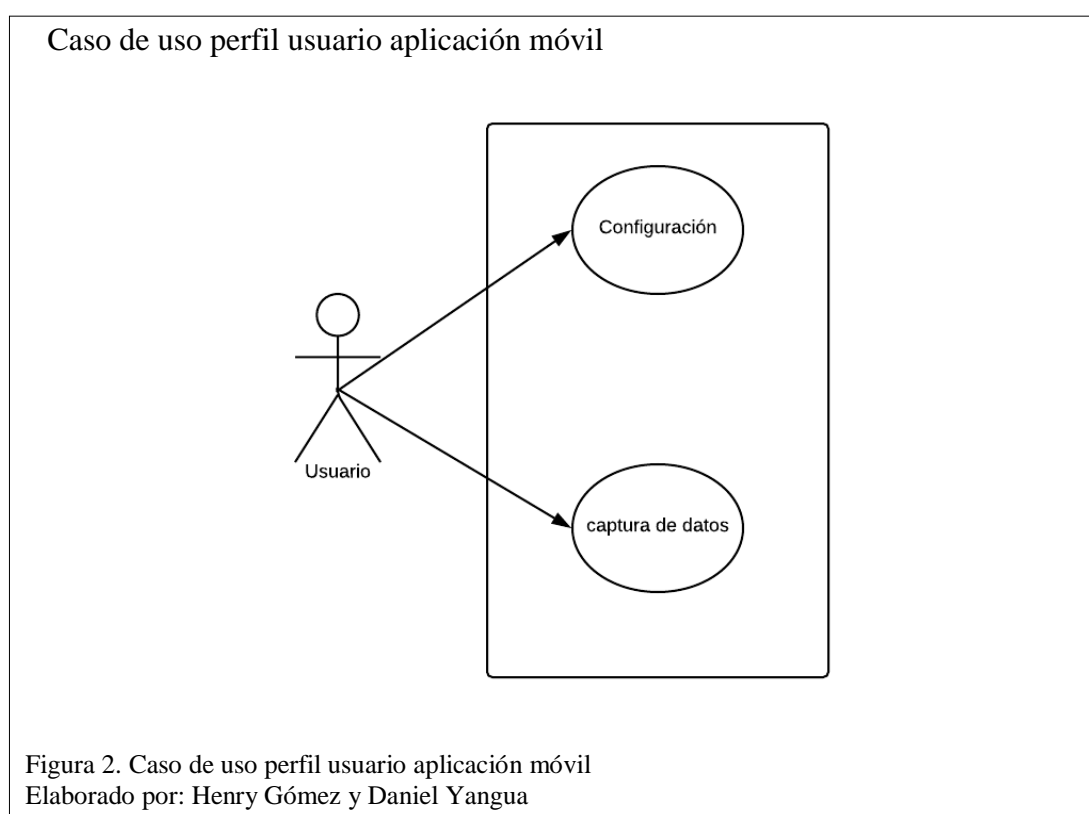
- La concurrencia debe ser máximo de 50 usuarios simultáneos al servidor, tanto en la aplicación móvil, como en la aplicación web.
- Optimizar el consumo de batería causado por el uso de aplicación móvil, en todo el trayecto en la carretera Quito – Riobamba.
- El consumo de memoria causado por el uso de aplicación móvil, debe ser inferior a 10 Mb (megabyte), en todo el trayecto en la carretera Quito – Riobamba.

## **2.2 Diagrama de casos de uso.**

Uno de los mayores desafíos de llevar a cabo este sistema, ha sido construir con componentes que no se conocen, un sistema totalmente nuevo, y es por ello que los casos de uso han sido totalmente imprescindibles para conocer la manera en la cual interactúan los usuarios con el sistema, hay que tomar en cuenta que uno de los

propósitos es el acceso libre a las aplicaciones, por tanto no debía existir ningún tipo de credenciales o formularios que el usuario debía rellenar, es por ello que los casos de uso se simplificaron, llegando en el caso de la aplicación web, a un sistema totalmente lineal en el que no hay opciones de configuración, únicamente la selección de los datos para llegar a la información final.

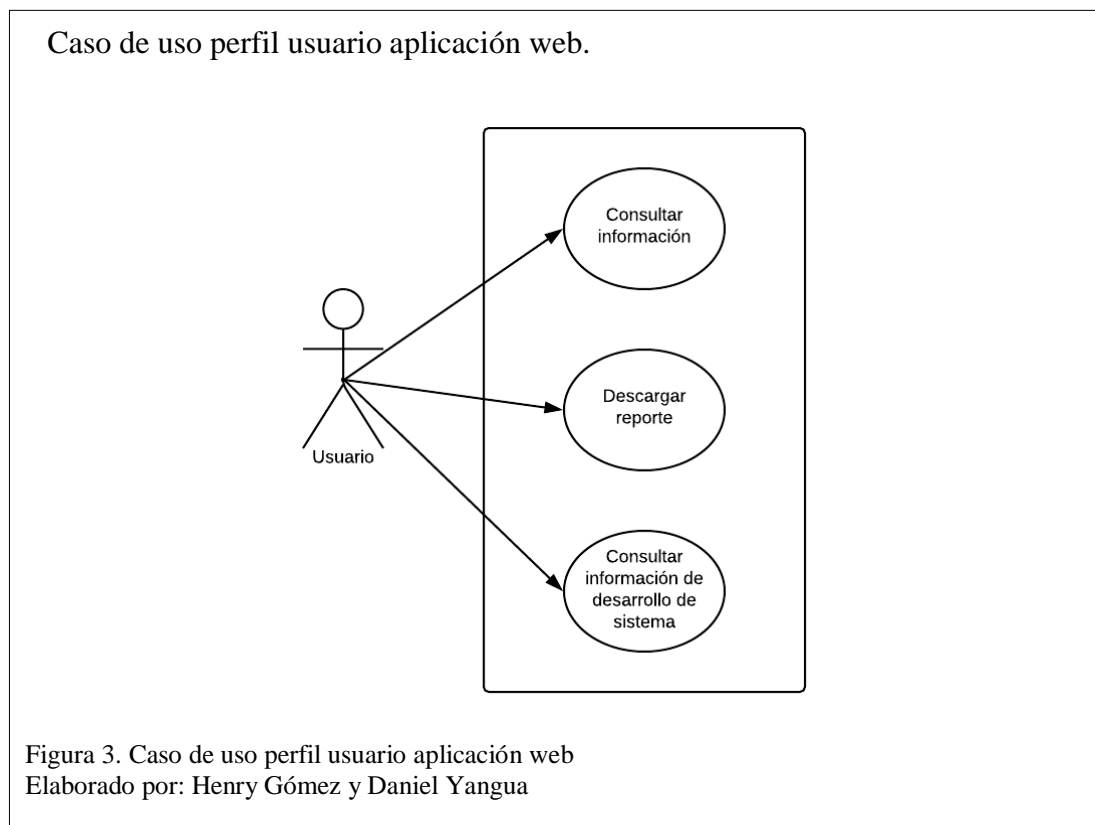
### 2.2.1 Caso de uso perfil usuario aplicación móvil.



En la figura No. 2, se observa como el usuario de la aplicación móvil tiene únicamente dos opciones de funcionamiento, las cuales son la configuración, y la captura de datos, esto se llevó a cabo así para que la interacción sea la menor posible, y de esta manera ser una aplicación más concisa en sus necesidades.



### 2.2.2 Caso de uso perfil usuario aplicación web.



En la figura No. 3, se observa los casos de uso de la aplicación web, en la que se observa que únicamente puede tener la opción de conocer la información acerca del desarrollo del sistema, consultar la información, y descargarla si es necesario, de esta manera se ha creado una aplicación totalmente libre de credenciales y de formularios de inscripción para consultar.

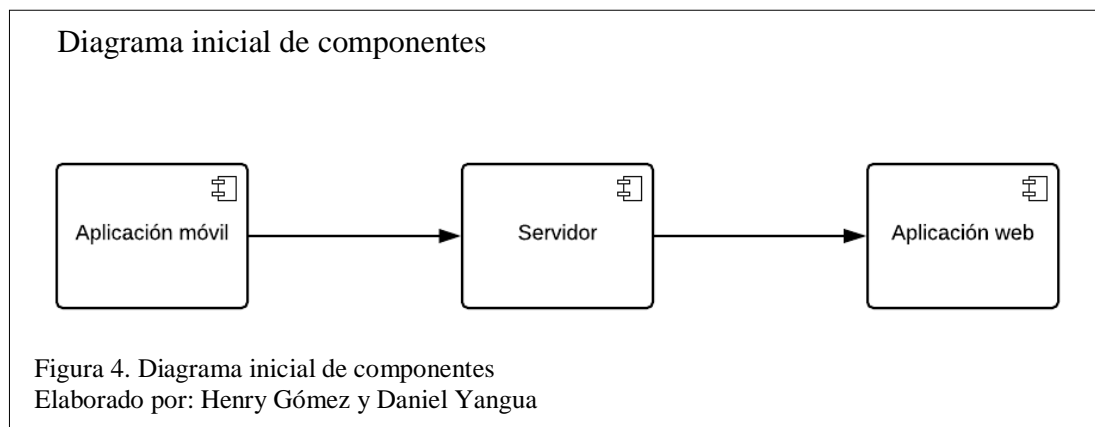
### 2.3 Análisis de arquitectura

La solución planteada para este proyecto, contiene los retos característicos desde el punto de vista funcional, la carga de datos obtenidos, así como el tratamiento de esta información, y la posterior forma en la que esta información se muestra, hace que la arquitectura del sistema sea un punto extremadamente relevante dentro del proyecto, es por ello que en primer lugar se llevó a cabo un análisis a profundidad de la arquitectura, y aunque ésta haya cambiado a lo largo del proyecto, no deja de ser una de las piedras angulares de dicho proyecto.

### 2.3.1 Arquitectura inicial.

Como punto de partida y debido a la definición de arquitectura planteada por numerosos autores, se debe llevar a cabo un análisis de cada uno de los componentes que interactuaran en el sistema, es por ello que en las primeras reuniones que se llevaron a cabo, el primer reto fue identificar estos componentes, y como estos debían interactuar entre sí, de esta manera se encontraron los siguientes componentes

- Aplicación móvil, con la cual se recolectarán los datos.
- Servidor, en el cual se procesará dicha información.
- Aplicación web, la cual mostrará los datos recolectados.

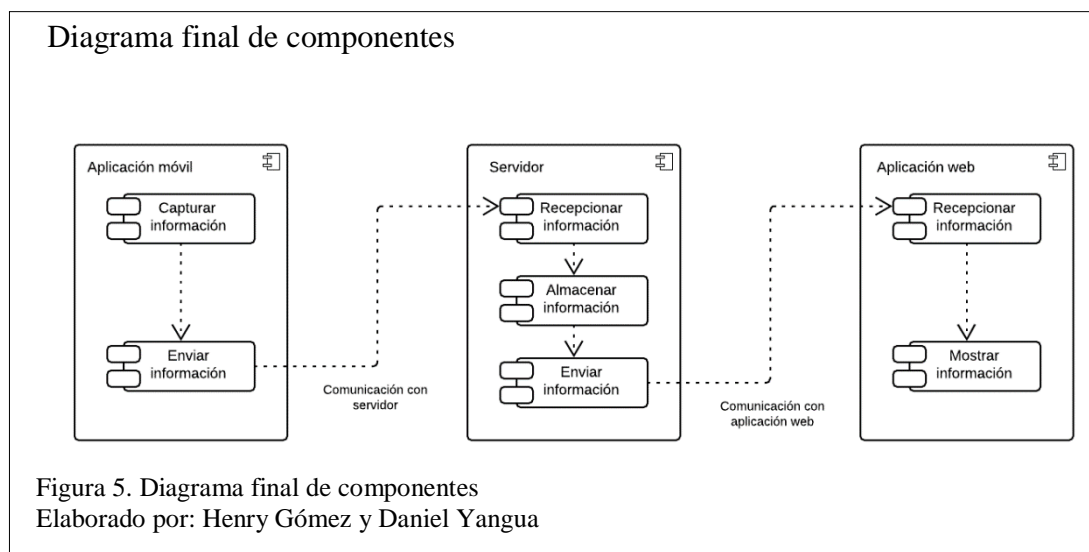


En la figura No. 4, se observa cómo estos componentes debían interactuar, siendo el servidor, el punto de conexión entre la aplicación móvil, y la aplicación web, aunque en aquel momento no se definían cómo se iba a interactuar funcionalmente, se tenía claro que el servidor era el encargado de recolectar, procesar y enviar la información para que ésta sea mostrada.

A partir de este momento se llevó a cabo un análisis más detallado de cada componente, y cómo debían interactuar entre sí, es por ello que, seleccionando las

tecnologías y estableciendo unos requerimientos iniciales, se pudo contemplar cómo debía ser esta arquitectura.

### 2.3.2 Diagrama de arquitectura planteado.

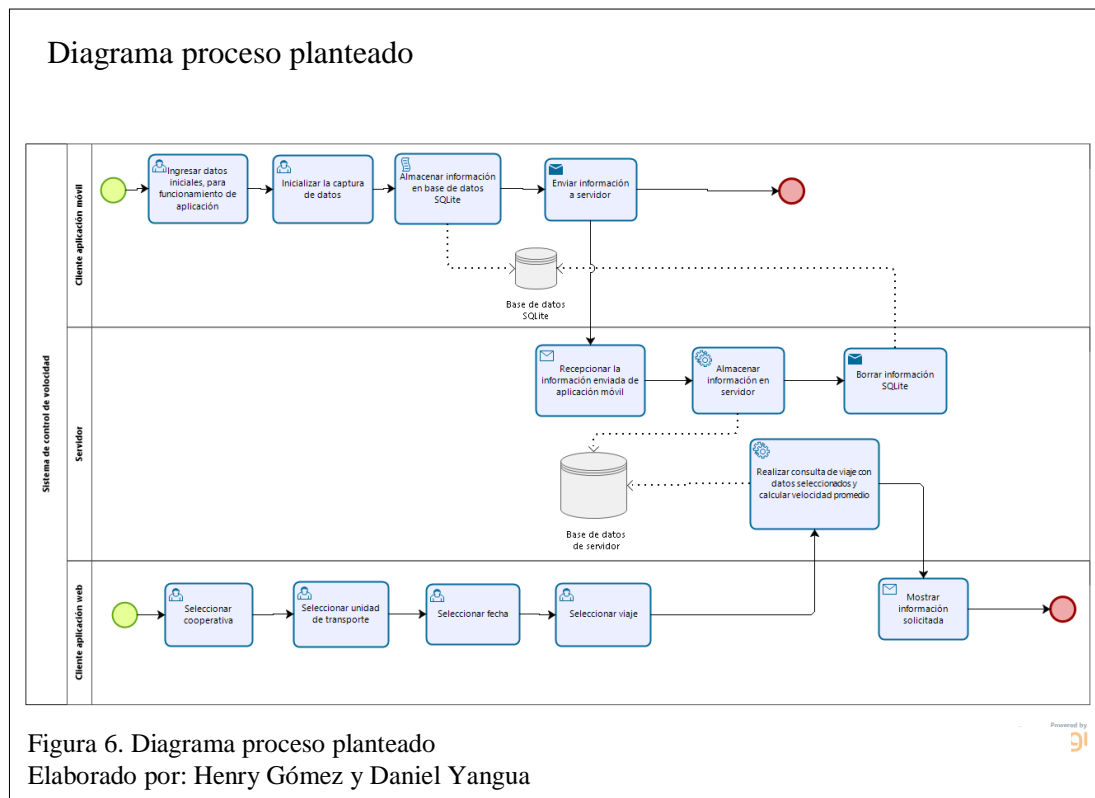


En la figura No. 5, se observa como aparecen nuevos componentes los cuales ayudan a mejorar la interacción de los componentes, estableciendo así una mejor comprensión del sistema, además que con la segmentación de los componentes es más sencillo encontrar cualquier tipo de fallos.

Al tener los servicios web por parte de la aplicación móvil, permite llevar a cabo una interacción tanto síncrona, como asíncrona de los datos obtenidos.

## 2.4 Análisis de proceso planteado.

Obtenido los requerimientos, así como el análisis de la arquitectura, se puede llevar a cabo el análisis del proceso, pero únicamente se puede presentar el proceso planteado de la aplicación, que anteriormente no existe un proceso definido para la captura de datos por parte de usuarios de las unidades de transporte, es por ello que únicamente se puede mostrar el análisis del proceso planteado.



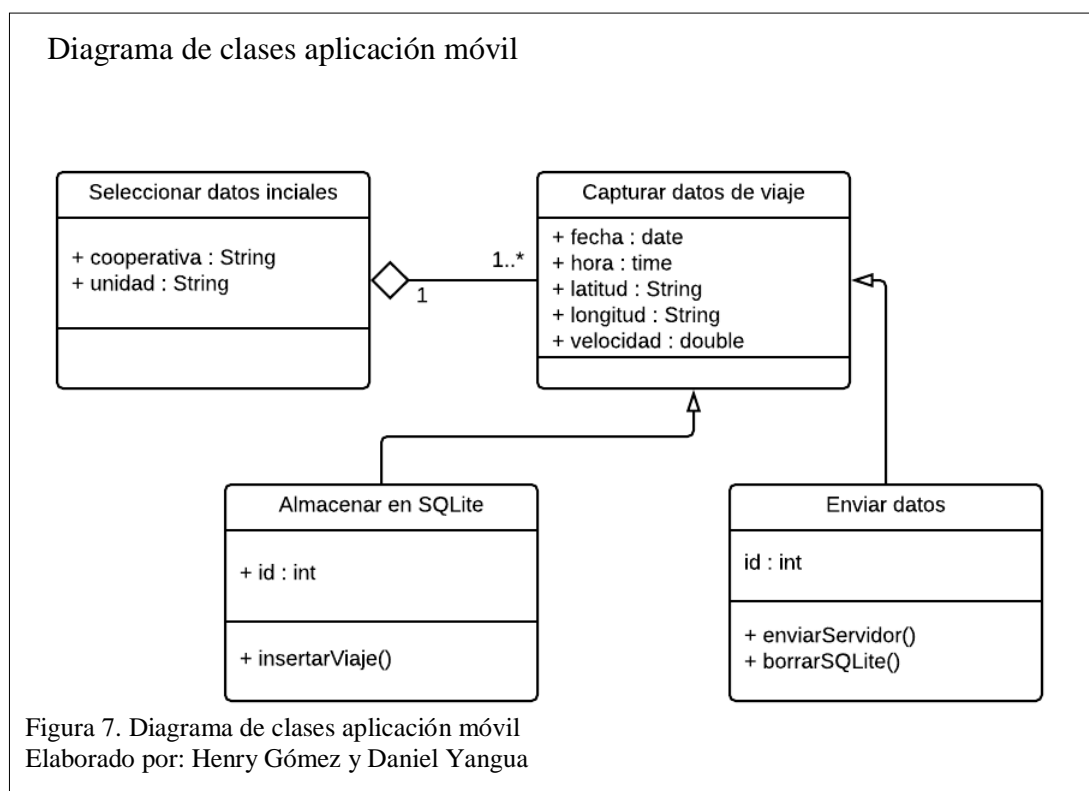
En la figura No. 6, se observa el proceso del sistema de control de velocidad, en el cual existe tres actores definidos, como son el usuario con la aplicación móvil, el servidor y el cliente con la aplicación web, de esta manera se puede observar como los diferentes actores interactúan entre sí, además de conocer más en profundidad los componentes de la aplicación, existiendo una aplicación de servidor la cual es la encargada de recibir la información enviada por parte del cliente de la aplicación

móvil, y enviar la orden de eliminar la información almacenada en la base de datos local.

Otro de los procesos importantes a destacar es el cálculo de la velocidad media, ya que existe la posibilidad de que en el vehículo se encuentre más de un usuario, y los GPS, de cada dispositivo de los usuarios pueden ser diferentes, puede existir una variación en los datos obtenidos, es por ello que se debe llevar a cabo un cálculo de la velocidad media, a la cual pasa el vehículo por el punto de latitud y longitud.

## 2.5 Diagrama de clases.

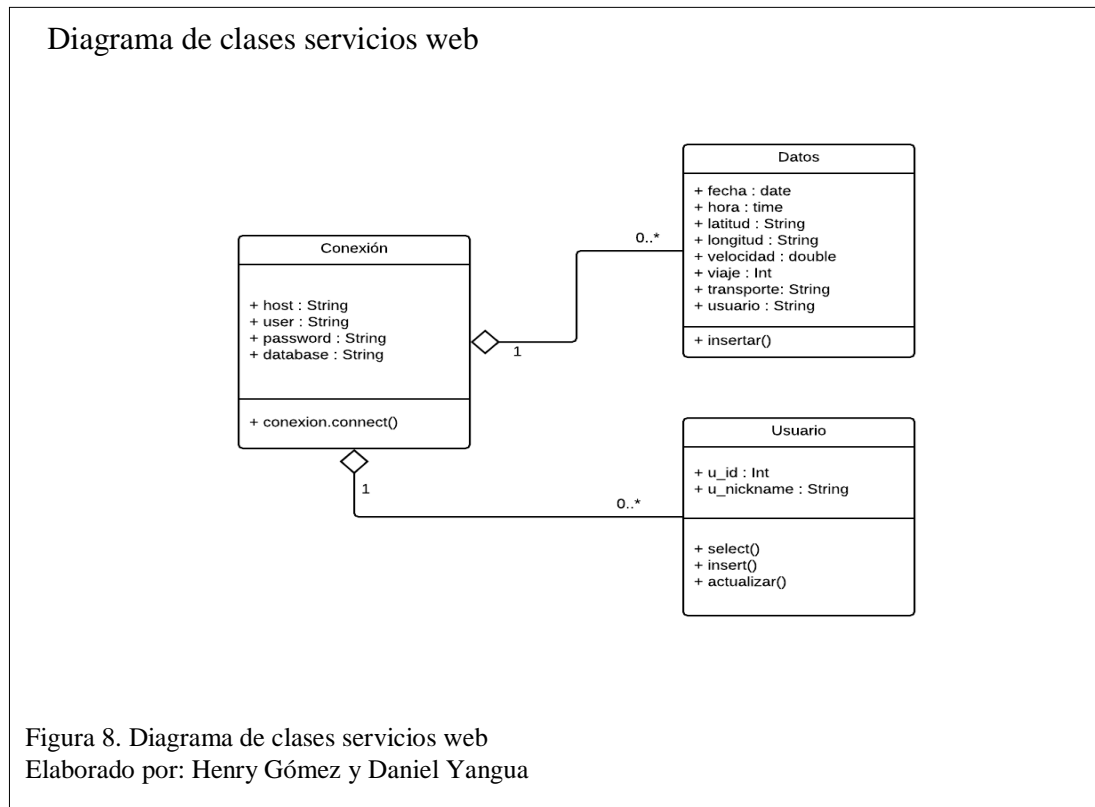
### 2.5.1 Diagrama de clases aplicación móvil.



En la figura No. 7, se observa la disposición de las clases de la aplicación móvil, y como realiza la captura de los datos, así como los almacena y los envía, hay que tomar en cuenta que se necesita un mecanismo en la aplicación móvil, para que no se

almacenen de forma permanente los datos que han sido enviados al servidor, y de esta manera obtener una aplicación más liviana.

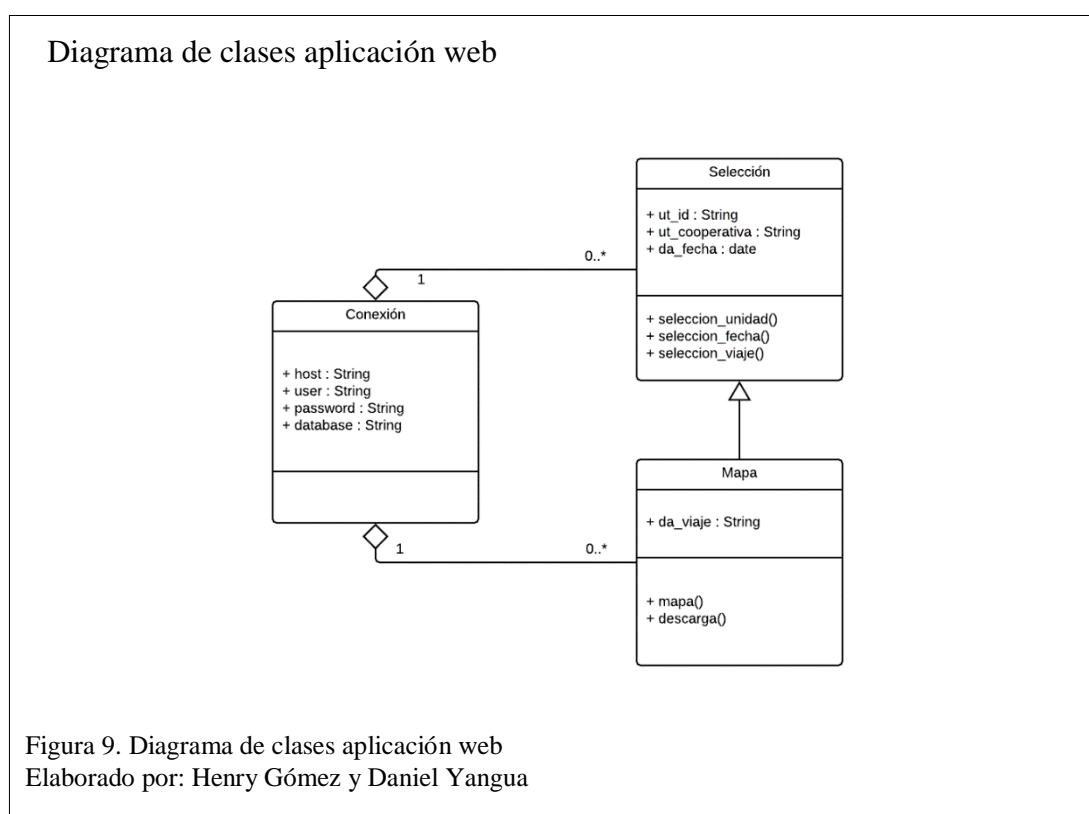
### 2.5.2 Diagrama de clases servicios web.



El diagrama de clases de servicios web está compuesto por tres clases como se observa en la figura No. 8, en las que se encuentra la clase conexión, la cual es la encargada de conectar con la base de datos que va a almacenar toda la información. Esta clase es indispensable para habilitar el funcionamiento de la aplicación, es por ello que las otras dos clases dependen de ésta.

En las otras dos clases se observa los métodos con los que se va a llevar a cabo las consultas, inserciones y actualizaciones, para después consultar desde la aplicación web.

### 2.5.3 Diagrama de clases aplicación web.



Como se observa en la figura No. 9, el diagrama de clases de aplicación web se divide en tres, al igual que el diagrama de clases de servicios web, es por ello que la clase conexión es imprescindible para las otras clases, ya que sin ella no se podrían conectar a la base de datos, y por tanto no se puede consultar esta información. Hay que recordar que esta aplicación es únicamente de consulta, no existe métodos de inserción, actualización, ni eliminación, además al ser una aplicación de open data, no se necesita ningún tipo de credenciales para acceder a ella.

Otra característica que se observa es la herencia que existe entre los datos de la clase selección hacia la clase mapa, esto se lleva a cabo de esta manera ya que existe una consulta secuencial hasta llegar al resultado para obtener la información en el mapa.

## 2.6 Diagrama de base de datos.

Después de obtener los requerimientos, las funcionalidades, y los casos de uso se define el diagrama de base de datos que se va a llevar a cabo, este diagrama será utilizado tanto en la aplicación web, como en la aplicación móvil, ya que esta base de datos va a contener toda la información que se capture desde la aplicación móvil, para luego ser consultada por la aplicación web.

Tabla 1.

Tablas de base de datos, versión final.

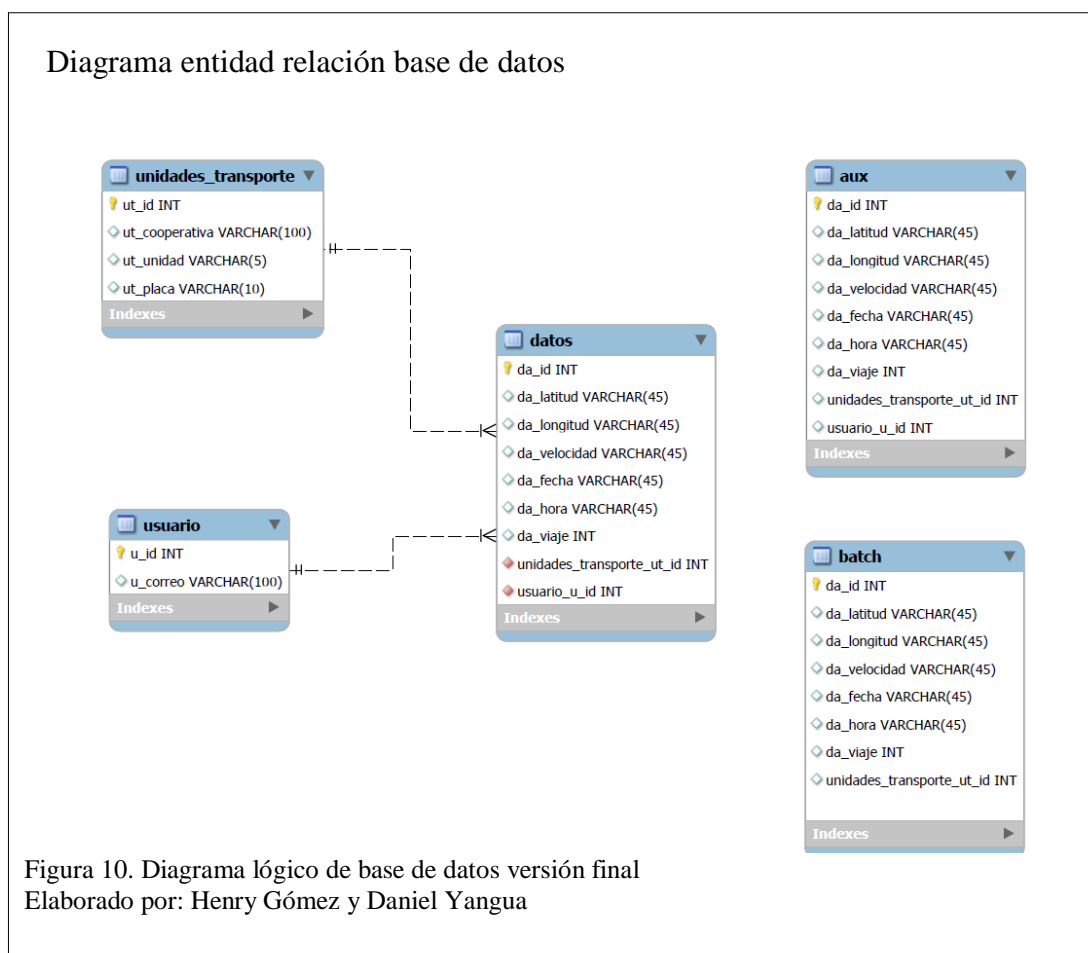
Tabla	Descripción
unidades_transporte	Almacena los datos de las unidades de transporte que realizan la ruta Quito – Riobamba.
datos	Almacena todos los datos que no han sido procesados.
aux	Almacena los datos con los que se va a llevar a cabo el promedio de la velocidad de cada punto que fue capturado, esta tabla se vacía una vez que se realice una nueva consulta
batch	Almacena la información final para el usuario de la aplicación web, esta información esta seleccionada y depurada.



usuario	Almacena la información de la persona que captura los datos, únicamente se solicita un nickname o alias.
---------	--

- Nota: Tabla de tablas de base de datos en su versión final.

### 2.6.1 Diagrama lógico base de datos, versión final.

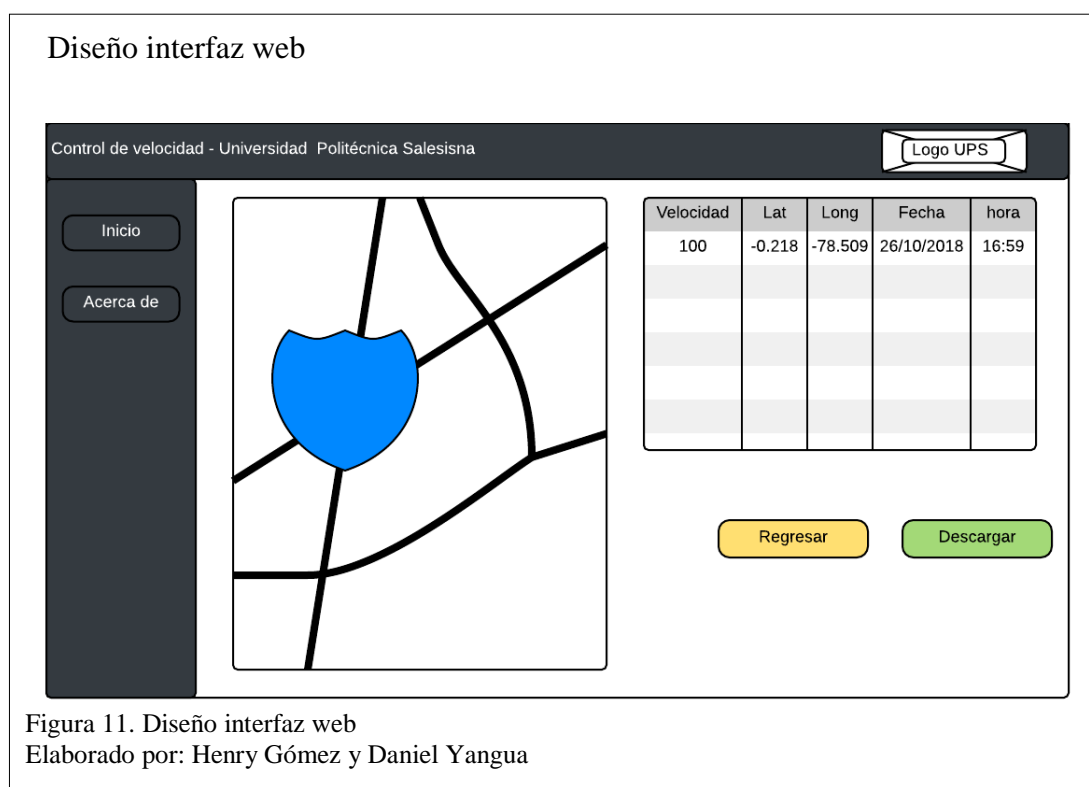


La figura No. 10, muestra el diagrama físico de la base de datos, en su versión final, con ella se cumple los requerimientos solicitados, y es la base de datos que interactúa con la aplicación web, y los servicios web.

## 2.7 Diseño de interfaces.

### 2.7.1 Diseño interfaz web.

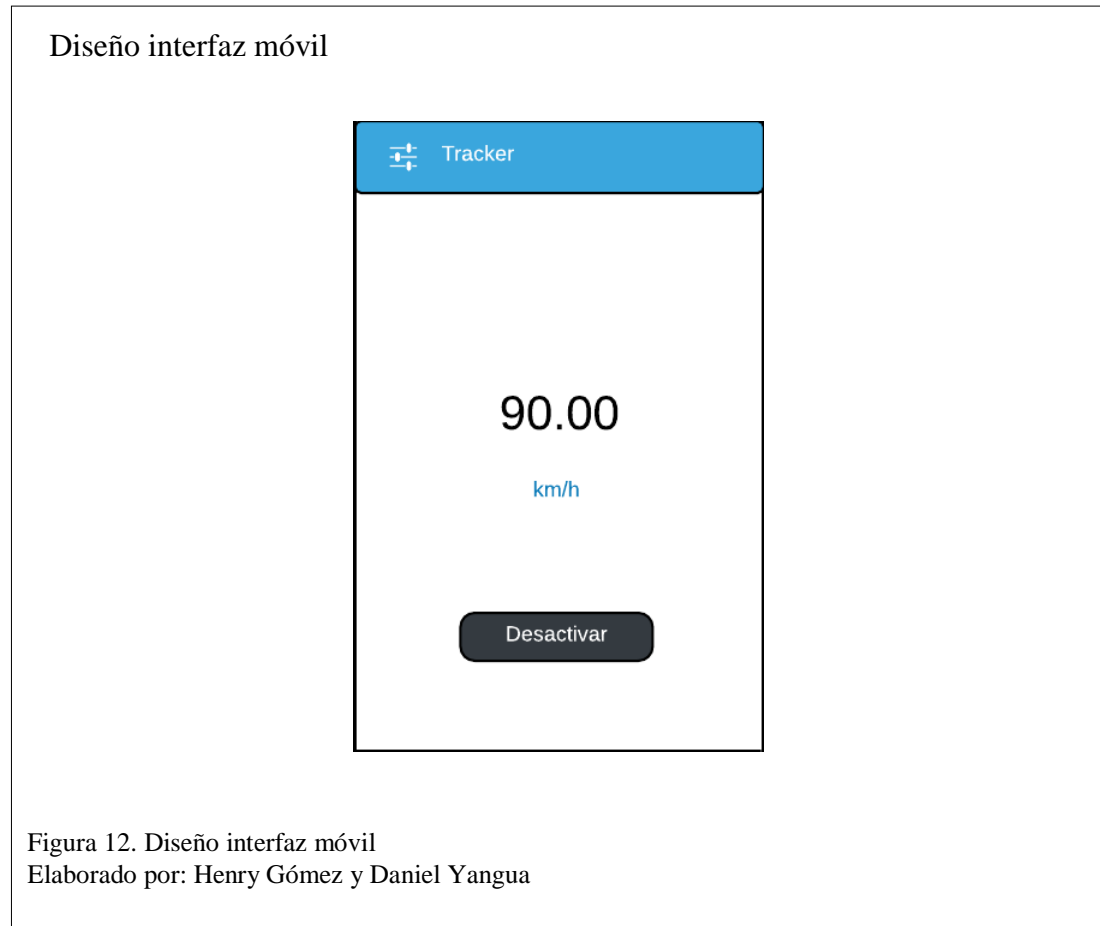
Para el diseño de interfaces se ha utilizado una plantilla destinada al uso de administración de sitios web, esta plantilla es SB Admin de uso gratuito y libre, la cual provee de todo lo necesario para la interfaz web, es por ello que el patrón de interfaz está establecido por el diseñador, aun así, se llevó a cabo la incrustación de un elemento ajeno a dicha plantilla la cual es el mapa en el que se mostraran las coordenadas de los puntos que se han seleccionado para mostrar.



Como se puede observar en la figura No. 11, se ha establecido el mapa dentro de la plantilla del diseñador.

### 2.7.1 Diseño interfaz móvil.

El diseño de la interfaz móvil, ha sido creado y ha seguido el patrón de colores de la aplicación web.



En la figura No. 12, se observa el diseño de interfaz de la aplicación móvil.

## Capítulo 3

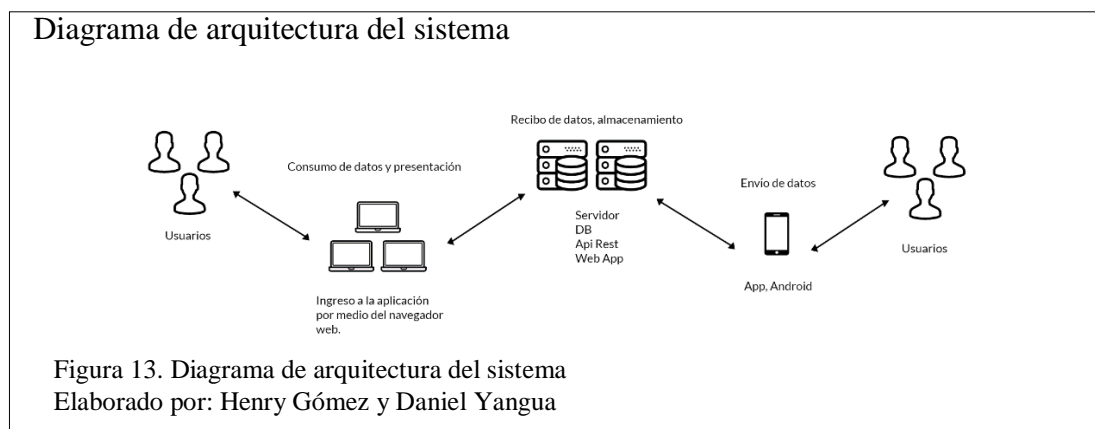
### Construcción y pruebas

#### 3.1 Arquitectura del sistema

El sistema está construido para la web sobre el patrón MVC (Modelo - Vista - Controlador) y para la aplicación móvil el modelo MVP (Modelo – Vista - Presentador) debido a que el framework de Android no presenta un claro desacoplamiento entre sus capas y comparte partes de lógica entre sus componentes, este patrón permite la rápida construcción de aplicaciones aislando los componentes tanto de la vista como la del modelo, el cual contiene la lógica de negocio, estos se comunican a través del controlador permitiendo así la abstracción y reutilización del código.

La aplicación web usa el framework CodeIgniter basado sobre el lenguaje de programación PHP.

Para la creación de los servicios se llevó a cabo la utilización de Express, un framework construido sobre JavaScript que permite la creación rápida de servicios web.

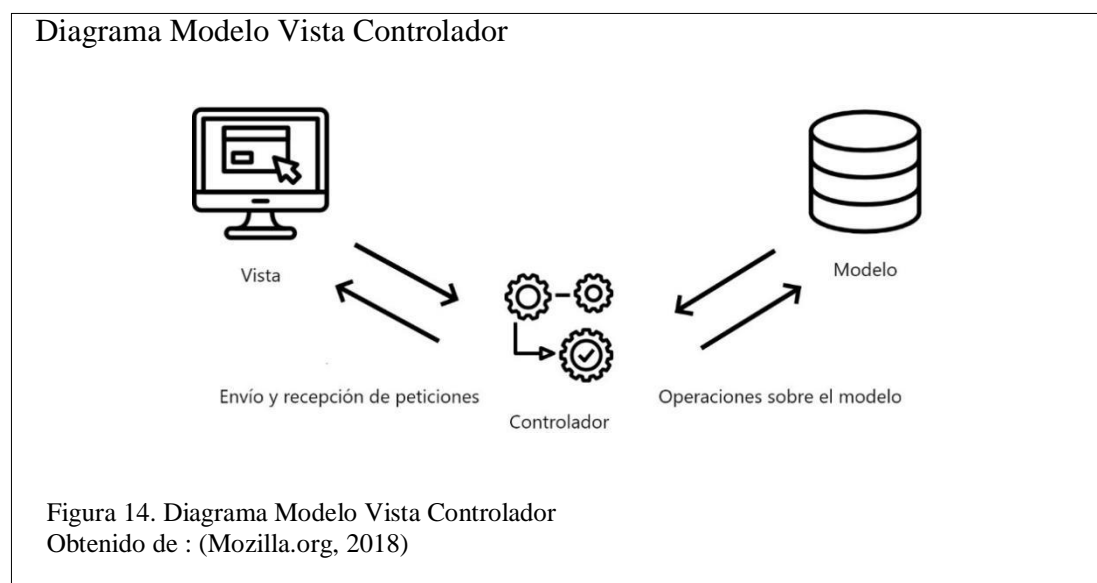


En la figura No. 13 se puede apreciar cómo funciona la solución empezando por la interacción del usuario con la aplicación móvil, la cual consta de 3 procesos importantes:

- Creación de un nickname o alias como nombre de usuario.
- Selección del Autobús
- Recolección de datos al iniciar un viaje.

La aplicación móvil es la encargada de recolectar los datos que luego serán enviados hasta el servidor, haciendo uso del estilo arquitectónico de REST, la información es enviada en un segundo plano de forma automática. Esta información llega hasta el servidor que se encarga de resolver dependiendo la petición que va a llevar a cabo, por ejemplo; el envío de un punto del mapa se hace a través de una petición Post el cual retorna un estado 201, este indica que la petición ha sido completada correctamente, esta información se puede consultar en la aplicación web, donde se encuentra diferentes acciones, las cuales puede realizar un usuario. Entre las principales consultar un viaje de acuerdo a la fecha y se muestra la información de ese viaje georreferenciado en el mapa.

### 3.2 Arquitectura de la aplicación web



La aplicación web está construida sobre el patrón MVC, con el uso de los principios de REST. Se hace uso de JavaScript asíncrono para evitar recargar la página al

momento de hacer consultas a la base de datos, la información se muestra de manera transparente para el usuario el cual puede realizar diferentes interacciones con la aplicación como consultar un viaje por fecha, filtrar los datos de las tablas, ver información en el mapa entre las principales, como se muestra en la figura No. 14.

### **3.2.1 Estándares Aplicación Web**

- Uso de guion bajo para nombrar variables por ejemplo `controller_user`.
- Nombres descriptivos para las funciones, clases y variables.
- Uso de etiquetas HTML de forma semántica.
- Evitar el uso de estilos en línea.
- Uso correcto de los verbos REST (get, post, put y delete).
- Separar los archivos por carpetas dependiendo el tipo.

### **3.2.2 Componentes del sistema web**

- Plantilla SB Admin
- Bootstrap 4
- Leaflet

### **3.2.3 Servicios Web**

Los servicios web están contruidos sobre Express.js un framework de Node.js que posee gran rendimiento al momento de gestionar gran cantidad de número de peticiones, elaborados sobre la filosofía REST haciendo uso correcto de los verbos como POST, PUT y GET, permite la comunicación entre la aplicación móvil y el servidor.

Ejemplo de servicio web para obtener las unidades de transporte

```
router.get('/', function(req, res, next) {  
  const myQuery = `SELECT * FROM unidades_transporte;`;   
  res.header("Access-Control-Allow-Origin", "*"); //security  
  db.query(myQuery, function (err, results, field) {  
    res.send(results);  
  });  
});
```

Figura 15. Ejemplo de servicio web para obtener las unidades de transporte  
Elaborado por: Henry Gómez y Daniel Yangua

En figura No. 15 se muestra un servicio web para obtener el listado de los buses almacenados en la base datos.

### 3.3 Arquitectura de la aplicación móvil

La aplicación móvil está construida sobre el patrón de diseño MVP (Modelo - Vista - Presentador) este patrón permite desacoplar la vista, se comunica a través de una interfaz el Presentador con la Vista y gestionar así los diferentes eventos que ocurren en la Vista.

#### 3.3.1 Estándares aplicación móvil.

Dentro de Android se acogen los estándares y convenciones del lenguaje de programación Java, así como sus buenas prácticas.

- Todos los nombres de clases inician con mayúsculas.
- Nombres descriptivos para las funciones, clases y variables.
- Constantes en mayúsculas por ejemplo MESSAGE = “mensaje de ejemplo”.
- Uso de nombres en singular de los archivos por ejemplo User.java.

#### 3.3.2 Componentes de la aplicación móvil

- Componentes del sistema operativo Android
- Retrofit 2
- SearchableSpinner

### 3.3.3 Permisos en AndroidManifest.xml

Es esencial administrar los permisos correctamente, este proceso es transparente para el usuario y es manejado por el framework de Android. Los permisos son declarados en el archivo AndroidManifest con extensión xml como se muestra en la figura No. 16, estos permiten añadir permisos según la aplicación los necesite, a partir de la versión veinte tres de Android los permisos se ejecutan en tiempo real y es necesario que el usuario apruebe los mismos para que la aplicación se pueda usar al 100%. (Google, 2018)



Los permisos necesarios para la aplicación son

- Acceso a la localización del GPS
- Acceso a internet
- Acceso al estado de la red



### 3.3.4 Declaración para la ejecución de servicios en AndroidManifest.xml

Los servicios en Android funcionan siempre que se los declare en el archivo AndroidManifest esto sucede como protección del sistema para que no se pueda ejecutar servicios en segundo plano sin que estén registrados en la aplicación. Provee mayor seguridad y además permite un mejor manejo de los servicios que se estén ejecutando en la aplicación.

#### Declaración de servicios en AndroidManifest.xml

```
<activity android:name=".MainActivity">
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

Figura 17. Declaración de servicios en AndroidManifest.xml  
Elaborado por: Henry Gómez y Daniel Yangua

En la figura No. 17 se muestra la declaración de servicio en el archivo AndroidManifest.xml, se debe indicar en la actividad que se ejecutará el servicio.

## 3.4 Código Relevante del Sistema

En este apartado se ha seleccionado porciones de código que muestran una posible solución para resolver un problema en particular.

### 3.4.1 Verificación de red WiFi para el envío de datos desde el dispositivo móvil

Este método permite verificar si el usuario tiene el WiFi de su equipo móvil encendido además si tiene conexión a internet, en ese caso ejecuta un método llamando sendAll() de la clase TrackerActions. Se puede apreciar dos mensajes en el entorno de desarrollo imprimiendo en la consola de Android Studio, en el cual dependiendo el estado si está encendido o apagado, este texto es usado solo como ejemplo y no se incluye en la aplicación.

El método `sendData` requiere como parámetro el `context` o contexto de la aplicación en cual se ejecutará, puede ser en primer o segundo plano, `ConnectivityManager` permite manejar distintos recursos dentro del sistema, `NetworkInfo` se encarga de devolver el estado del recurso al cual se realice la petición en este caso al recurso del WiFi

Método de verificación de red WiFi para el envío de datos.

```
public void sendData(Context context) {  
    ConnectivityManager manager = (ConnectivityManager) getSystemService(context.CONNECTIVITY_SERVICE);  
    NetworkInfo networkInfo = manager.getNetworkInfo(ConnectivityManager.TYPE_WIFI);  
    TrackerActions trackerActions = new TrackerActions(getApplicationContext());  
  
    if (networkInfo.isConnected()) {  
        trackerActions.sendAll();  
        Log.i(TAG, "sendData: WIFI ON");  
    } else {  
        Log.i(TAG, "sendData: WIFI OFF");  
        trackerActions.close();  
    }  
}
```

Figura 18. Método de verificación de red WiFi para el envío de datos.  
Elaborado por: Henry Gómez y Daniel Yangua

Como se muestra en la figura No. 18, se realiza el proceso en segundo plano, de forma que el usuario no tenga que preocuparse por encender o apagar el recurso mediante una interacción con la interfaz, esto permite que esté receptando en lapsos de tiempo determinado o siempre esté activo, mejora la experiencia del usuario y tiene un bajo costo en el uso de la batería.

### 3.4.2 Uso del GPS del equipo móvil

Los equipos celulares cuentan con GPS que varía dependiendo el costo del equipo, puede funcionar con internet usando servicios de Google o sin conexión a internet, en la aplicación móvil se usa sin conexión a internet para evitar el uso de datos móviles del usuario, se puede utilizar los dos servicios al mismo tiempo.

### Implementación de la clase LocationListener

```
public class LocationListenerC implements LocationListener {  
    private Context context;  
    private Intent intent;  
  
    public LocationListenerC(Context context) {  
        this.context = context;  
    }  
  
    @Override  
    public void onLocationChanged(Location location) {  
        // code  
    }  
  
    @Override  
    public void onStatusChanged(String s, int i, Bundle bundle) {  
        // code  
    }  
  
    @Override  
    public void onProviderEnabled(String s) {  
        // code  
    }  
  
    @Override  
    public void onProviderDisabled(String s) {  
        // code  
    }  
}
```

Figura 19. Implementación de la clase LocationListener  
Elaborado por: Henry Gómez y Daniel Yangua

Esta clase implementa distintos métodos como se muestra figura No. 19, se pueden usar dependiendo las acciones que se realicen en el GPS, el principal método es `onLocationChanged` este método escucha cada vez que la posición del GPS cambie, se puede configurar para que detecte por parámetros de distancia o tiempo, los otros métodos son de apoyo para saber cuándo el GPS fue activado o desactivado. En este caso se implementó un proceso en segundo plano que ejecute la tarea de escuchar al GPS y guarde los datos en una base de datos local SQLite para luego detectar si el dispositivo móvil tiene acceso a internet y enviar la información.

### 3.4.3 Implementación de un servicio con GPS

Se implementa un servicio para realizar una tarea en segundo plano, se crea una instancia de la clase `locationManager` para manejar el servicio de localización, éste permite administrar el tipo de proveedor a usar, se debe pasar dos parámetros muy importantes que es el tipo y la distancia en la cual la clase `LocationListener` escuchará los eventos del GPS, si se deja en cero los dos valores como se puede apreciar significa que estará escuchando a cualquier cambio de localización o de tiempo.



Como se aprecia en la figura No. 20, la clase `GpsService` permite asignar un evento que inicie o finalice el uso del GPS, es muy importante permitir al usuario finalizar el servicio cuando éste lo considere necesario, debido a que el uso de este recurso afecta la duración de la batería causando una pésima experiencia para el usuario. `LocationListener` debe cancelar a la suscripción cuando el servicio sea finalizado, de

igual forma cuando la aplicación se cierre, se puede observar ese proceso en el método `OnDestroy` que se ejecuta cuando el servicio esté en estado de stop (detenido).

#### 3.4.4 Cálculo de la velocidad entre dos distancias con el GPS del móvil

La velocidad se puede calcular de distintas formas haciendo uso de funciones matemáticas o de métodos que brinda el framework de Android. Para disminuir el error en el cálculo se usó un método de la clase `LocationListener` llamado `distanceTo`, esta función recibe dos parámetros que son dos objetos de tipo `location` que contienen la información como latitud, longitud, fecha, etc. El método retorna la distancia en metros sobre la cual se operó con cálculo del tiempo que contiene el objeto `location`.

##### Líneas de código del cálculo de la velocidad

```
float distanceTo = (startLocation.distanceTo(endLocation) / 1000);

float hoursBetweenToDates = formatDate.HoursBetweenToDates(
    formatDate.formatTime(currentTime),
    formatDate.formatTime(lastTime));

String speed = calculateSpeed(distanceTo, hoursBetweenToDates);
```

Figura 21. Líneas de código del cálculo de la velocidad  
Elaborado por: Henry Gómez y Daniel Yangua

Se puede apreciar en la figura No. 21 la transformación del tiempo, la fecha que usa por defecto el objeto `location` es de tipo `Long` que puede ser convertido a distintos formatos haciendo uso de la clase `Date` que permite obtener la fecha en distintos formatos. El cálculo final sucede en el método `calculateSpeed` el cual recibe dos parámetros la distancia obtenida como `float` del método `distanceTo` y el tiempo transcurrido entre las dos fechas. El uso de `distanceTo` reduce el riesgo de trabajar con decimales y devuelve la aproximación más cercana al valor real sobre el cual se puede operar para obtener la velocidad en ese punto georreferenciado.

### 3.4.5 Envío de datos de forma asíncrona desde el móvil

Para este proceso se utilizó la librería Retrofit2 para el envío de peticiones entre la aplicación móvil y servicios web, un punto importante a tener en cuenta es el envío de información respecto a los viajes de un usuario, al tener una cantidad representativa de datos podría afectar el rendimiento de la aplicación móvil, manejar este proceso de forma síncrona causa el problema de tener que esperar que se envíe una petición para que otra pueda iniciar el proceso, al hacer uso de un proceso asíncrono se puede hacer envíos sucesivos sin que se produzca ese tiempo de espera entre el envío de una petición y otra.

Envío de datos de manera asíncrona al servidor.

```
public void sendDataServer(TrackerPojo trackerPojo) {
    Gson gson = new GsonBuilder()
        .setLenient()
        .create();

    Retrofit.Builder builder = new Retrofit.Builder()
        .baseUrl(UrlHelper.URL)
        .addConverterFactory(GsonConverterFactory.create(gson));

    Retrofit retrofit = builder.build();
    PointClient client = retrofit.create(PointClient.class);
    Call<String> call = client.sendPoint(trackerPojo);

    call.enqueue(new Callback<String>() {
        @Override
        public void onResponse(Call<String> call, Response<String> response) {
            if(response.code() == 200 || response.code() == 201) {
                deleteTrack(id);
            }
        }
        @Override
        public void onFailure(Call<String> call, Throwable t) {
            Log.i("****", "onResponse: " + t.toString());
        }
    });
}
```

Figura 22. Envío de datos de manera asíncrona al servidor  
Elaborado por Henry Gómez y Daniel Yangua

Como se muestra en la figura No. 22 se lleva a cabo un patrón de desarrollo de software que consiste en ejecutar una función dentro de otra función, esto es posible gracias a que Retrofit2 usa la librería RxJava.

### 3.4.6 Cálculo de la media de las velocidades aplicativo Web

Una parte importante es permitir que los usuarios obtengan tiempos más cortos de respuesta en las consultas al obtener un gran volumen de datos entre los viajes. Se creó una función que permita calcular la velocidad media de las velocidades y al mismo tiempo se almacene en una tabla de la base de datos para evitar volver hacer el proceso. La próxima vez simplemente el usuario que requiere esta información consulta a la base datos, mejorando la experiencia del usuario con un tiempo de respuesta inferior, Esta función se muestra en la figura No. 23.

Función del cálculo medio de las velocidades.

```
public function mapa($ut_id, $da_fecha, $da_viaje){
    $sql0 = "DELETE FROM aux";
    if ($da_viaje == "---- Seleccione ----"){
        $da_viaje = 0;
    }
    $query0 = $this->db->query($sql0);
    $sql1 = "SELECT * FROM datos WHERE unidades_transporte_ut_id = ".$ut_id." AND da_fecha = '".$da_fecha."' AND da_viaje = ".$da_viaje."";
    $query1 = $this->db->query($sql1);
    foreach ($query1->result() as $row) {
        $u_id = $row->usuario_u_id;
        $hora = $row->da_hora;
        $sql1 = "SELECT DISTINCT da_fecha, da_hora, AVG(da_velocidad) as da_velocidad, AVG(da_latitud) as da_latitud, AVG(da_longitud) as da_longitud from datos
        WHERE unidades_transporte_ut_id = ".$ut_id." AND da_fecha = '".$da_fecha."' AND da_viaje = ".$da_viaje." AND da_hora = '".$hora."";
        $query1 = $this->db->query($sql1);
        foreach ($query1->result() as $row2) {
            $velocidad = $row2->da_velocidad;
            $latitud = $row2->da_latitud;
            $longitud = $row2->da_longitud;
        }
        if ($velocidad >= 90 && $velocidad <= 180){
            $sql2 = "INSERT INTO aux VALUES ('".$latitud."', '".$longitud."', '".$da_fecha."', '".$hora."', ".$velocidad.", ".$da_viaje.", ".$ut_id.", ".$u_id.")";
            $query2 = $this->db->query($sql2);
        }
    }
    $sql3 = "SELECT DISTINCT da_velocidad, da_latitud, da_longitud, da_fecha, da_hora FROM aux";
    $query3 = $this->db->query($sql3);
    foreach ($query3->result() as $row3) {
        $la = $row3->da_latitud;
        $lo = $row3->da_longitud;
        $ve = $row3->da_velocidad;
        $ho = $row3->da_hora;
        $sql4 = "INSERT into batch VALUES (0, '".$la."', '".$lo."', '".$da_fecha."', '".$ho."', ".$ve.", ".$da_viaje.", ".$ut_id.")";
        $query4 = $this->db->query($sql4);
    }
    if ($query3->num_rows() > 0){
        return $query3;
    }else{
        return FALSE;
    }
}
```

Figura 23. Función del cálculo medio de las velocidades  
Elaborado por: Henry Gómez y Daniel Yangua

Porción de código para cargar el mapa con los datos del viaje correspondiente.

```
<?php
if ($mapa != "") {
    foreach ($mapa->result() as $row) {
        echo "<tr>";
        echo "<td>".$row->da_velocidad."</td>";
        echo "<td>".$row->da_latitud."</td>";
        echo "<td>".$row->da_longitud."</td>";
        echo "<td>".$row->da_fecha."</td>";
        echo "<td>".$row->da_hora."</td>";
        echo "</tr>";
    }
}
?>
```

Figura 24. Porción de código para cargar el mapa con los datos del viaje correspondiente  
Elaborado por: Henry Gómez y Daniel Yangua

Para los procesos que muestran la información, se usa el método antes mencionado junto con otros métodos menos relevantes como se muestra en la figura No. 24.

### 3.5 Pruebas

Las pruebas en el sistema permiten comprobar que el funcionamiento de cada aplicativo funcione correctamente de manera individual y conjunta, además de someter a distintos escenarios y ver la manera en que se desempeña.

#### 3.5.1 Implementación de las aplicaciones para el escenario de pruebas

Para el escenario de pruebas se implementó las aplicaciones en servidores de pocos recursos de manera que permita llevar a cabo el plan de pruebas.

Características del servidor de la aplicación web

- PHP, versión 5.6
- MySQL, versión 10.0
- Apache, versión 2.4
- Arquitectura x86\_64
- Sistema operativo Linux
- Apache versión 2.4



- 6GB de capacidad

Para los servicios web se usó Heroku que es una plataforma que brinda servicios para alojar aplicaciones de distintos lenguajes de programación en nuestro caso Node.js con una capacidad de 512 MB RAM.

### **3.5.2 Plan de pruebas**

Para el plan de pruebas se seleccionaron distintos tipos de pruebas con el objetivo de comparar cada aplicativo sometido ante distintos escenarios y como el usuario interactúa con la misma.

- Pruebas unitarias
- Pruebas de rendimiento
- Pruebas de estrés
- Pruebas de funcionalidad
- Pruebas de usabilidad

### **3.5.3 Pruebas Unitarias**

Permiten verificar el correcto funcionamiento de los servicios web, se lleva a cabo pruebas específicas de cada caso, las cuales son programadas y se espera un resultado en concreto. Para estas pruebas se usó dos librerías construidas sobre el lenguaje Javascript que permite probar la funcionalidad de cada servicio web en diferentes escenarios.

### 3.5.3.1 Obtener la lista de unidades de transporte

Test unitarios lista de unidades de transporte

```
1 let chai = require('chai');
2 let chaiHttp = require('chai-http');
3 const expect = require('chai').expect;
4
5 chai.use(chaiHttp);
6 const url = 'https://dyangua.herokuapp.com';
7
8 describe('get buses: ', () => {
9   it('should get all buses', (done) => {
10     chai.request(url)
11       .get('/buses')
12       .end(function (err, res) {
13         console.log(res.body)
14         expect(res).to.have.status(201);
15         done();
16       });
17   });
18 });
```

Figura 25. Test unitarios lista de unidades de transporte  
Elaborado por: Henry Gómez y Daniel Yangua

En el primer escenario se simula obtener la lista completa de unidades de transporte este proceso lo realiza el aplicativo móvil para obtener la lista de transporte previamente insertados en la base de datos como se muestra en la figura No. 25.

Resultado obtenido del test

```
get buses:
[ { ut_id: 1,
  ut_cooperativa: 'Andina',
  ut_unidad: '001',
  ut_placa: 'CCC-0000' },
  { ut_id: 2,
    ut_cooperativa: 'Ecuador Ejecutivo',
    ut_unidad: '001',
    ut_placa: 'CCC-0000' },
  { ut_id: 3,
    ut_cooperativa: 'Patria',
    ut_unidad: '001',
    ut_placa: 'CCC-0000' },
  { ut_id: 4,
    ut_cooperativa: 'Riobamba',
    ut_unidad: '001',
    ut_placa: 'CCC-0000' },
  { ut_id: 5,
    ut_cooperativa: 'Transvencedores',
    ut_unidad: '001',
    ut_placa: 'CCC-0000' } ]
✓ should get all buses (850ms)

1 passing (860ms)
```

Figura 26. Resultado obtenido del test  
Elaborado por: Henry Gómez y Daniel Yangua

Se obtiene un tiempo de 850 ms de respuesta del servidor, al momento de obtener la lista de usuarios como se muestra en la figura No. 26

### 3.5.2.2 Verificar la existencia de un usuario

Verificación de la disponibilidad de nombre de usuario (alias)

```
let chai = require('chai');
let chaiHttp = require('chai-http');
const expect = require('chai').expect;

chai.use(chaiHttp);
const url = 'https://dyangua.herokuapp.com';

describe('Insert user: ', () => {
  it('should create a new user', (done) => {
    chai
      .request(url)
      .get("/usuario?nickname=dario")
      .end(function(err, res) {
        console.log(res.body);
        expect(res).to.have.status(200);
        done();
      });
  });
});
```

Figura 27. Verificación de la disponibilidad de nombre de usuario (alias)  
Elaborado por: Henry Gómez y Daniel Yangua

Este servicio permite verificar que exista un usuario único para posteriormente insertar la información a la base de datos como se muestra en la figura No. 27.

Resultado obtenido del test de verificación de usuarios.

```
Insert user:
Usuario ya existe
✓ should create a new user (593ms)

1 passing (603ms)
```

Figura 28. Resultado obtenido del test de verificación de usuarios  
Elaborado por: Henry Gómez y Daniel Yangua

Al momento de verificar la existencia de un usuario se obtiene un tiempo de respuesta 593 ms por parte del servidor como se muestra en la figura No. 28.

### 3.5.2.3 Verificación de la creación de punto de georreferenciado

Verificación de la creación de punto de georreferenciado

```
const expect = require("chai").expect;

chai.use(chaiHttp);
const url = "https://dyangua.herokuapp.com";

describe("insert point: ", () => {
  it("should insert a point", done => {
    chai
      .request(url)
      .post("/punto")
      .send({
        "lat" : "1",
        "lon" : "1",
        "vel" : "1",
        "fecha" : "26/05/2018",
        "hora" : "21:45",
        "viaje" : "1",
        "transporte" : "1",
        "usuario" : "2"
      })
      .end(function(err, res) {
        console.log(res.body);
        expect(res).to.have.status(200);
        done();
      });
  });
});
```

Figura 29. Verificación de la creación de punto de georreferenciado  
Elaborado por: Henry Gómez y Daniel Yangua

Este test permite verificar la inserción de un nuevo punto georreferenciado en la base de datos, este contiene toda la información necesaria para ser mostrado en la aplicación web cuando se realiza una consulta de un viaje como se muestra en la figura No. 29.

### Resultado obtenido verificación de la creación de punto de georreferenciado

```
insert point:
{}
✓ should insert a point (587ms)

1 passing (601ms)
```

Figura 30. Resultado obtenido verificación de la creación de punto de georreferenciado  
Elaborado por: Henry Gómez y Daniel Yangua

Los resultados obtenidos en la inserción de un nuevo registro que contiene la información de georreferenciación fue un tiempo de respuesta de 601 ms por parte del servidor como se muestra en la figura No. 30.

#### 3.5.2.4 Ejemplo de test esperando un error

##### Ejemplo de test esperando un error

```
chai.use(chaiHttp);
const url = "https://dyangua.herokuapp.com";

describe("get error point: ", () => {
  it("should get a error point", done => {
    chai
      .request(url)
      .get("/punto")
      .send({
        "lat": "1",
        "lon": "1",
        "vel": "1",
        "fecha": "26/05/2018",
        "hora": "21:45",
        "viaje": "1",
        "transporte": "1",
      })
      .end(function (err, res) {
        console.log(res.body);
        expect(res).to.have.status(400);
        done();
      });
  });
});
```

Figura 31. Ejemplo de test esperando un error  
Elaborado por: Henry Gómez y Daniel Yangua

Como se aprecia en la figura No.31 el test fue creado con el fin de mostrar la utilidad de los test unitarios en una aplicación, consta de no enviar un parámetro en la petición POST esperando un error 400 que significa que la petición está mal realizada.

### 3.5.3 Pruebas de Rendimiento

Estas pruebas se realizaron usando el software JMeter creado por Apache Software Foundation, el objetivo es poner a prueba el sistema bajo un número de peticiones establecidas para cada escenario en el cual su objetivo se centra en verificar el funcionamiento del sistema para detectar y mitigar posibles tiempos altos de espera que existan en las aplicaciones.

#### - Escenario 1

Este escenario contempla un usuario en el sistema en un tiempo de 1s, realizando peticiones HTTP Request.

Tabla 2.

Prueba de rendimiento con carga de un usuario (primera parte)

Etiqueta	# Peticiones	Media	Max	Min	Error (%)	Desviación
HTTP Request	1	488 ms	488 ms	488 ms	0.00%	0.00

- Nota: La tabla muestra los resultados de tiempos obtenidos del escenario 1.

Como se observa en la tabla 2 se obtiene los resultados probando un usuario en el sistema, el resultado de la media puede variar por lo cual es importante tomar en cuenta la desviación al momento de analizar los datos.

Tabla 3.

Prueba de rendimiento con carga de un usuario (segunda parte)

# Peticiones	Rendimiento	Recibidos Kb/sec	Enviados Kb/sec	Media Bytes
1	204.918	1.38	0.26	691.0

- Nota: La tabla muestra los resultados de rendimiento obtenidos del escenario 1.

Como se observa en la tabla 3, el rendimiento obtenido en el escenario es de 204.918 este número representa el desempeño del servidor y éste indica que el servidor está en condiciones aptas de soportar el número de usuarios procesados.

- **Escenario 2**

Este escenario contempla 50 usuarios en el sistema en un tiempo de 1s, realizando peticiones HTTP Request

Tabla 4.

Prueba de rendimiento con carga de 50 usuarios (primera parte)

Etiqueta	# Peticiones	Media	Max	Min	Error (%)	Desviación
HTTP Request	50	3386 ms	417 ms	6369 ms	0.00%	1761.64

- Nota: La tabla muestra los resultados de tiempos obtenidos del escenario 2.

Como se observa en la tabla 4, el servidor maneja 50 usuarios se obtiene un error de 0.00% con tiempos de respuestas superiores al escenario 1, estos datos se acercan mucho más a la realidad en cuanto al número de usuarios esperados para el sistema.

Tabla 5.

Prueba de rendimiento con carga de 50 usuarios (segunda parte)

# Peticiones	Rendimiento	Recibidos Kb/sec	Enviados Kb/sec	Media Bytes
1	677.782	4.57	0.86	691.0

- Nota: La tabla muestra los resultados de rendimiento obtenidos del escenario 2

Como se observa en la tabla 5, el rendimiento del servidor se ve afectado considerablemente teniendo en cuenta que en el escenario uno se obtiene valores lejos de la realidad en los cuales el sistema no solo va a recibir un usuario, este escenario es más ideal en cuanto al flujo de usuarios que se espera tener en la aplicación.

- **Escenario 3**

Este escenario contempla 100 usuarios en el sistema en un tiempo de 1s, realizando peticiones HTTP Request.

Tabla 6.

Prueba de rendimiento con carga de 100 usuarios (primera parte)

Etiqueta	# Peticiones	Media	Max	Min	Error (%)	Desviación
HTTP Request	100	6924 ms	401 ms	13460 ms	0.00%	3807.96

- Nota: La tabla muestra los resultados de tiempos obtenidos del escenario 3.

Como se observa en la tabla 6, este escenario es como se espera que el sistema escale para obtener un flujo de 100 usuarios en 1 segundo, interactuando en las aplicaciones web y móvil, el número de respuesta mínimo aumenta considerablemente y aun así el servidor sigue obteniendo unos tiempos de respuesta bajos.

Tabla 7.

Prueba de rendimiento con carga de 100 usuarios (segunda parte)

# Peticiones	Rendimiento	Recibidos Kb/sec	Enviados Kb/sec	Media Bytes
1	677.782	4.57	0.86	691.0

- Nota: La tabla muestra los resultados de rendimiento obtenidos del escenario 3.

Como se observa en la tabla 7, el rendimiento del servidor se ve afectado considerablemente, pero siendo capaz de manejar 100 usuarios de forma correcta.



- **Escenario 4**

Este escenario contempla 250 usuarios en el sistema en un tiempo de 5s, realizando peticiones HTTP Request.

Tabla 8.

Prueba de rendimiento con carga de 250 usuarios (primera parte)

Etiqueta	# Peticiones	Media	Max	Min	Error (%)	Desviación
HTTP Request	250	15561 ms	406 ms	30216 ms	2.20%	8784.13.96

- Nota: La tabla muestra los resultados de tiempos obtenidos del escenario 4.

Como se observa en la tabla 8, los tiempos de repuesta casi desbordan el límite de la capacidad del servidor, se obtiene un tiempo mínimo de respuesta bastante alto y el tiempo máximo se mantiene, esto sucede por la forma de gestionar las peticiones en el servidor, hay que tomar en cuenta que el tiempo mínimo es el tiempo representativo que un usuario usando la aplicación esperaría para obtener el recurso esperado.

Tabla 9.

Prueba de rendimiento con carga de 250 usuarios (segunda parte)

# Peticiones	Rendimiento	Recibidos Kb/sec	Enviados Kb/sec	Media Bytes
1	709.99	4.79	0.90	691.5

- Nota: La tabla muestra los resultados de rendimiento obtenidos del escenario 4.

Como se observa en la tabla 9, el rendimiento del servidor se ve afectado considerablemente ante este punto, los tiempos de respuesta son mayores lo cual podría producir problemas de escalabilidad en el sistema además de no poder resolver la solicitud de recursos para el 2.2% de usuarios que significa 7 usuarios de 250.

### - **Análisis de los Resultados**

En los cuatro escenarios se describen distintas medidas para simular la carga en el sistema, obteniendo resultados de como respondería el servidor ante cada uno. En el primer escenario se obtiene los mejores resultados, pero este no es el número de usuarios esperados para el sistema, el escenario 2 es el ideal en el cual se espera tener 50 usuarios por segundo interactuando con el sistema. En el escenario 3 el rendimiento del servidor se mantiene, pero en el escenario 4 se termina llegando a los límites de usuario del sistema.

#### **3.5.4 Pruebas de Estrés**

Para las pruebas de estrés se trabaja con los datos anteriormente obtenidos en las pruebas de rendimiento.

Tabla 10.

Datos de escenarios para el análisis de las pruebas de estrés

Escenario	# Peticiones	Rendimiento	Error
1	1	204.918	0.00%
2	50	677.782	0.00%
3	100	689.703	0.00%
4	250	709.099	2.20%

- Nota: La tabla muestra los resultados de los escenarios de las pruebas de rendimiento, para el análisis en las pruebas de rendimiento.

Dentro de los 3 primeros escenarios como se aprecia en la tabla 10, el sistema responde de una forma correcta, no tiene problemas con una concurrencia de 100 usuarios, obteniendo tiempos de respuesta bajos para garantizar el tiempo de envío y recepción de recursos. En el cuarto escenario se ve afectado el rendimiento del servidor a partir de los 250 usuarios, el sistema tiende a fallar con un error 2.20%. El sistema se espera

que trabaje dentro del segundo escenario escalando hasta el tercer escenario, el cuarto escenario está muy lejos de cumplirse debido a la popularidad e impacto que tenga el sistema en el caso de que esto se presente, el sistema puede crecer en hardware. Actualmente está limitado por la cantidad de recursos del servidor.

Es importante recalcar que las pruebas de estrés permiten tener una idea del número de usuarios que soportara el sistema antes de fallar, esto se puede mitigar de dos formas dependiendo los casos, en el primer caso refactorizando código, en el segundo caso es mejorar las características del servidor teniendo en cuenta que es necesario que se debe trabajar en los dos casos conforme la exigencia del sistema crezca.

### **3.5.5 Pruebas de Funcionalidad**

Estas pruebas se centran en comprobar que las características y funcionalidades de cada aplicativo del sistema, se realizan los procesos que el usuario lleva a cabo dentro del sistema.

#### **3.5.5.1 Pruebas de funcionalidad en la aplicación web.**

La tabla consta de cinco parámetros

- #: Referencia al número de prueba
- Prueba: Proceso al cual fue sometido la prueba
- Acciones: Eventos realizados para poder llevar a cabo la prueba
- Estado: Se manejó dos estados completado e incompleto dependiendo el resultado de la prueba.
- Observaciones: Se describe si existió algún tipo de incidente al realizar la prueba.

Tabla 11.  
Pruebas funcionales de la aplicación web

#	Prueba	Acciones	Estado	Observaciones
1	Ingreso a la aplicación web	Acceder a la aplicación web mediante el navegador.	Completado	
2	Seleccionar una cooperativa de transporte	Seleccionar en una lista el nombre de la cooperativa.	Completado	
3	Seleccionar cantidad de registros a mostrar	Seleccionar la cantidad de registros de cooperativas a desplegarse en la tabla.	Completado	
4	Buscar una cooperativa	Usar el cuadro de texto para buscar para filtrar las cooperativas por nombres	Completado	
5	Botón Continuar	Avanzar después de seleccionar una cooperativa el botón continuar permite pasar a seleccionar una	Completado	

		unidad de transporte caso contrario no lo permite hasta seleccionar una cooperativa.		
6	Seleccionar una unidad de transporte	Seleccionar en una lista el nombre de una unidad de transporte este corresponde solo a las unidades que corresponden a la cooperativa seleccionada.	Completado	
7	Seleccionar cantidad de registros a mostrar	Seleccionar la cantidad de registros de unidades de transporte a desplegarse en la tabla.	Completado	
8	Buscar una cooperativa	Usar el cuadro de texto para buscar para filtrar las unidades de transporte por nombres	Completado	
9	Botón Continuar	Avanzar después de seleccionar una unidad de	Completado	

		transporte el botón continuar permite pasar a seleccionar la fecha entre una lista correspondiente a los viajes en caso de no seleccionar una unidad de transporte el botón no le permite avanzar en el proceso.		
10	Botón regresar	Permitir regresar al proceso anterior, la selección una cooperativa.	Completado	
11	Seleccionar una fecha correspondiente a un viaje	Avanzar después de seleccionar una fecha correspondiente a todos los viajes realizados sobre los parámetros antes seleccionados.	Completado	
12	Seleccionar cantidad de registros a mostrar	Seleccionar la cantidad de registros de viajes a	Completado	

		desplegarse en la tabla.		
13	Buscar una fecha	Usar el cuadro de texto para buscar para filtrar los viajes realizados correspondientes a la fecha insertada.	Completado	
14	Botón Continuar	Avanzar después de seleccionar una fecha correspondiente a un viaje se procede a mostrar la información en el mapa en caso de que no se ha seleccionado una fecha el botón no le permite continuar.	Completado	
15	Botón regresar	Permitir regresar al proceso anterior, la selección una unidad de transporte.	Completado	
16	Seleccionar un número de viaje.	Seleccionar en una lista una los números de	Completado	

		viajes que posee esa unidad de transporte en la fecha seleccionada.		
17	Botón Continuar	Avanzar después de seleccionar un viaje, el botón continuar permite pasar a mostrar en el mapa los datos correspondiente a esa búsqueda.	Completado	
18	Botón regresar	Permitir regresar al proceso anterior, la selección de una fecha.	Completado	
19	Zoom acercar (mapa)	Permitir acercar la distancia focal del mapa.	Completado	
20	Zoom alejar (mapa)	Permitir alejar la distancia focal del mapa.	Completado	
21	Marcador o punto	Mostrar la información correspondiente a ese georreferenciado	Completado	



22	Seleccionar cantidad de registros a mostrar	Seleccionar la cantidad de registros de puntos georreferenciados a desplegarse en la tabla.	Completado	
23	Buscar un numero de viaje	Usar el cuadro de texto para buscar para filtrar los datos basado en el criterio de la velocidad.	Completado	
24	Botón regresar	Permitir regresar al proceso anterior, la selección de un viaje.	Completado	
25	Botón Descargar XLS	Permitir descargar los datos en una hoja de calcula de Excel.	Completado	
26	Menú de navegación	Mostrar el icono a lo largo de aplicación de manera de facilitar al usuario regresar hasta un proceso sin tener que seguir de manera	Completado	

		secuencial el orden de acceso de los procesos.		
--	--	--	--	--

- Nota: La tabla hace referencia a las pruebas funcionales de la aplicación web.

### 3.5.5.2 Pruebas de funcionalidad de la aplicación móvil.

- #: Referencia al número de prueba
- Prueba: Proceso al cual fue sometido la prueba
- Acciones: Eventos realizados para poder llevar a cabo la prueba
- Estado: Se manejó dos estados completado e incompleto dependiendo el resultado de la prueba.
- Observaciones: Se describe si existió algún tipo de incidente al realizar la prueba.

Tabla 12.  
Pruebas funcionales de la aplicación móvil

#	Prueba	Acciones	Estado	Observaciones
1	Ingresar un nuevo alias	Registrar un nuevo alias para asignar un identificador al usuario	Completado	Este alias es simbólico no se precisa datos adicionales por que no es tomado como usuario.
2	Actualizar el alias	Modificar el nombre de alias antes ingresado	Completado	

3	Guardar (vista usuario)	Permitir almacenar o modificar el alias	Completado	El alias al momento de almacenar se asigna un número correspondiente a un id dentro del sistema.
4	Cargar de cooperativas de transporte	Almacenar de manera local la lista que contiene los nombres de las cooperativas de transporte.	Completado	Este proceso es transparente para para el usuario, se ejecuta en una tarea por detrás cuando se inicia la aplicación.
5	Carga de unidades de transporte	Almacenar de manera local la lista que contiene los nombres de las unidades de transporte.	Completado	Este proceso es transparente para para el usuario, se ejecuta en una tarea por detrás cuando se inicia la aplicación
6	Actualizar lista de cooperativas de transporte	Actualizar la lista de cooperativas de transporte en la base local del móvil.	Completado	Este proceso es transparente para para el usuario, se ejecuta en una tarea por detrás

				cuando se inicia la aplicación
7	Actualizar lista de unidades de transporte	Actualizar la lista de unidades de transporte en la base local del móvil.	Completado	Este proceso es transparente para para el usuario, se ejecuta en una tarea por detrás cuando se inicia la aplicación
8	Botón Cooperativas	Permite abrir un cuadro de dialogo.	Completado	
9	Cuadro de dialogo de cooperativas	Mostar la lista de cooperativas disponibles para seleccionar.	Completado	
10	Buscar cooperativa	Filtrar la lista de cooperativas mediante el parámetro de búsqueda del nombre	Completado	
11	Botón Close (cuadro de dialogo de cooperativas)	Permitir cerrar el cuadro de dialogo	Completado	

12	Botón Unidades	Permite abrir un cuadro de dialogo.	Completado	
13	Cuadro de diálogo de unidades de transporte	Mostar la lista de unidades de transporte disponibles para seleccionar.	Completado	
14	Buscar unidad de transporte	Filtrar la lista de unidades de transporte mediante el parámetro de búsqueda del nombre	Completado	
15	Botón Close (cuadro de dialogo de unidades de transporte)	Permitir cerrar el cuadro de dialogo	Completado	
16	Botón Empezar viaje	Permitir dirigirse a la vista de empezar viaje.	Completado	Este botón permite redirigir de manera automática al usuario para que pueda empezar a guardar sus viajes no obstante es importante que se haya seleccionado

				una unidad de transporte y exista un usuario para que se habilite.
17	Botón Activar	Permitir iniciar los procesos para empezar almacenar los datos viaje	Completado	
18	Botón Desactivar	Detiene los procesos de almacenar los datos de viaje.	Completado	
18	Envío de datos recolectados.	Enviar al servidor los datos del viaje almacenados en el equipo móvil	Completado	Este proceso se ejecuta en segundo plano de manera automática cada que el usuario abra la aplicación el proceso se ejecutado dentro de un rango de tiempo verifica si existen datos almacenados posteriormente una vez

				enviados son eliminados.
20	Botón Menú	Mostrar la lista de vistas disponibles para la navegación en la aplicación.	Completado	
21	Botón X (cerrar menú)	Ocultar la barra de menú.	Completado	Este botón se muestra siempre que el menú este activo.
22	Opción menú usuario	Dirigir hacia la vista de usuario.	Completado	
23	Opción seleccionar autobús	Dirigir hacia la vista de seleccionar autobús.	Completado	
24	Opción seleccionar salir	Cierra la aplicación.	Completado	

- Nota: La tabla hace referencia a las pruebas funcionales de la aplicación móvil.

### 3.5.6 Pruebas de usabilidad

Se realizó una encuesta con el fin de obtener retroalimentación de como el usuario percibe el sistema, probar cada una de las características de las aplicaciones y encontrar posibles incongruencias en la interfaz de usuario, que puedan causar una mala experiencia. En la encuesta participaron 10 personas para probar el sistema, el rango de edades consta entre 20 – 45 años.

## - Encuesta aplicación web

### Resultado de la pregunta 1, encuesta aplicación web.

¿ Cómo calificaría su experiencia en el uso de la aplicación ?

10 respuestas

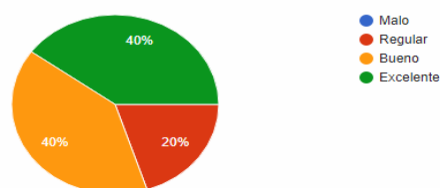


Figura 32. Resultado de la pregunta 1, encuesta aplicación web  
Elaborado por: Henry Gómez y Daniel Yangua

Se puede apreciar en la figura No. 32 que el 80% de los usuarios tiene experiencia entre buena y excelente y solo un 20% lo ve como regular, ningún usuario lo percibido como malo demuestra que el sistema está diseñado para brindar una buena experiencia al usuario.

### Resultado de la pregunta 2, encuesta aplicación web.

¿ Los elementos usados en el aplicativo como botones, cuadros de texto, etiquetas, etc. Le resultaron familiares ?

10 respuestas

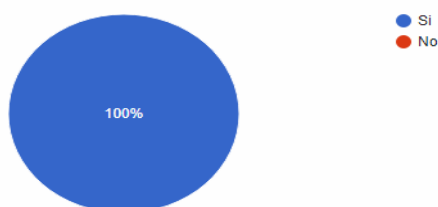


Figura 33. Resultado de la pregunta 2, encuesta aplicación web.  
Elaborado por: Henry Gómez y Daniel Yangua

Como se demuestra en la figura No. 33 todos los usuarios demuestran que se siente familiarizados con elementos usados en la aplicación web, es muy importante que un botón se vea como lo que es, y de esa manera no confundir al usuario.



Resultado de la pregunta 3, encuesta aplicación web.

¿ Las ayudas proporcionadas en cada proceso del aplicativo le resulto útil ?

10 respuestas

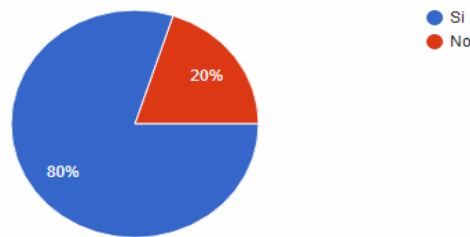


Figura 34. Resultado de la pregunta 3, encuesta aplicación web.  
Elaborado por: Henry Gómez y Daniel Yangua

Como se muestra en la figura No. 34, en la aplicación web se brindó un guía en cada proceso para ayudar a los usuarios nuevos, puede ser muy efectivo y al mismo tiempo molestar a usuarios más experimentados. En este caso se lo acepto bien porque sirve de guía en caso de no saber cómo interactuar con la aplicación.

Resultado de la pregunta 4, encuesta aplicación web.

¿ Considera apropiado el diseño de la aplicación (colores, fuente, posición de la información, etc) ?

10 respuestas

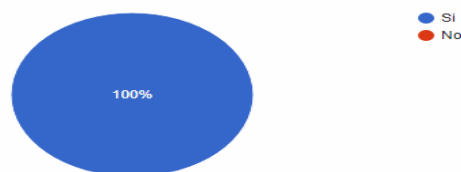
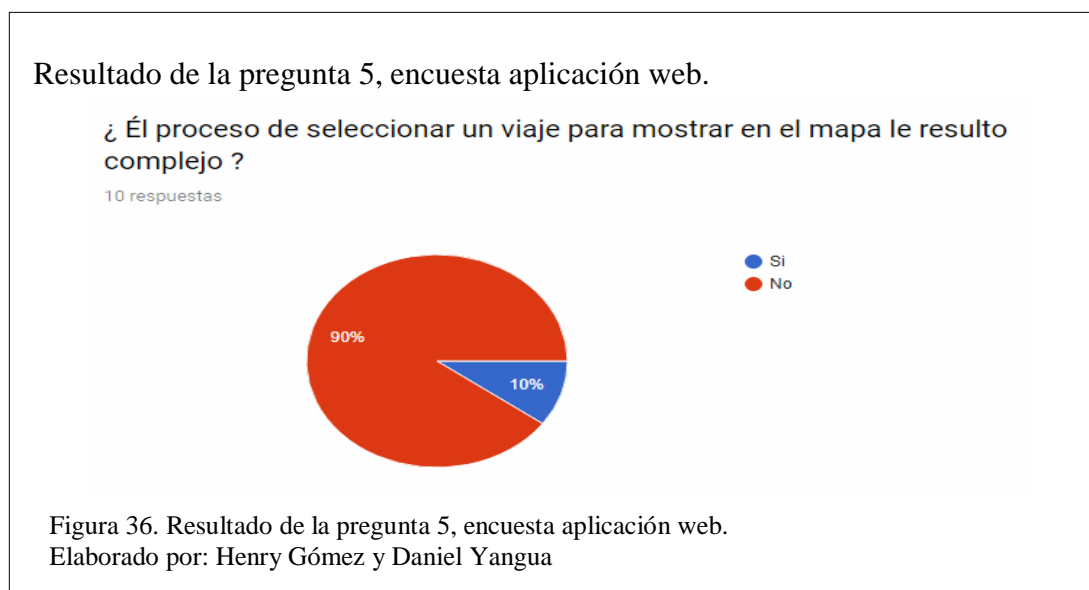


Figura 35. Resultado de la pregunta 4, encuesta aplicación web.  
Elaborado por: Henry Gómez y Daniel Yangua

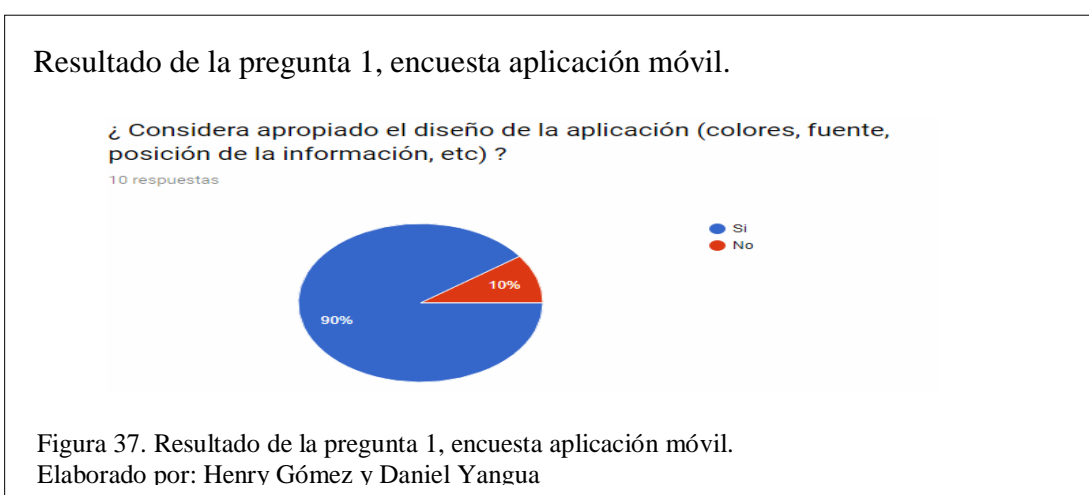
Como se muestra en la figura No. 35 la comprensión del diseño es esencial, en una aplicación se debe usar la fuente adecuada, con los colores que contrasten con la otra aplicación del sistema, que no sean intrusivos y la organización de la información se

muestre correctamente en la aplicación web, todos los usuarios lo encontraron apropiado.

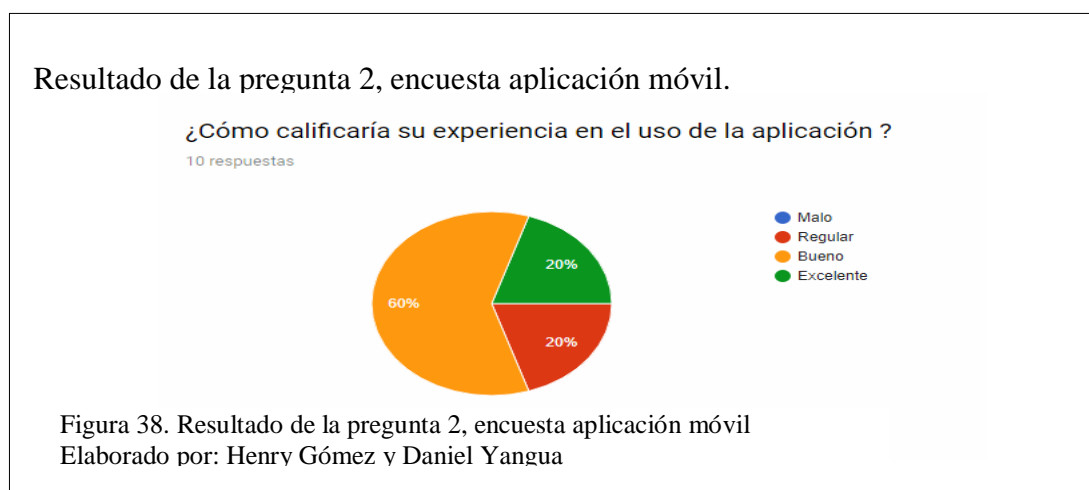


Como se muestra en la figura No.36, este es el proceso más importante de la aplicación web para que el usuario puede ingresar y después de completar las condiciones de búsqueda pueda llegar a observar el mapa con la información deseada, solo un usuario tuvo problemas con el proceso esto indica que el 90% de los usuarios pueden lograr cumplir todo el proceso sin ningún problema.

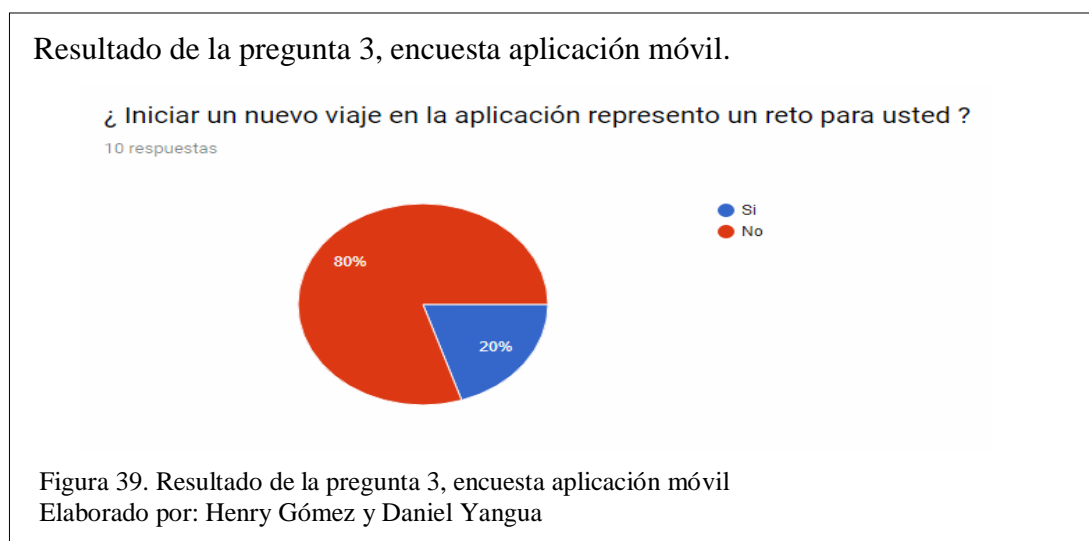
#### - Encuesta aplicación móvil



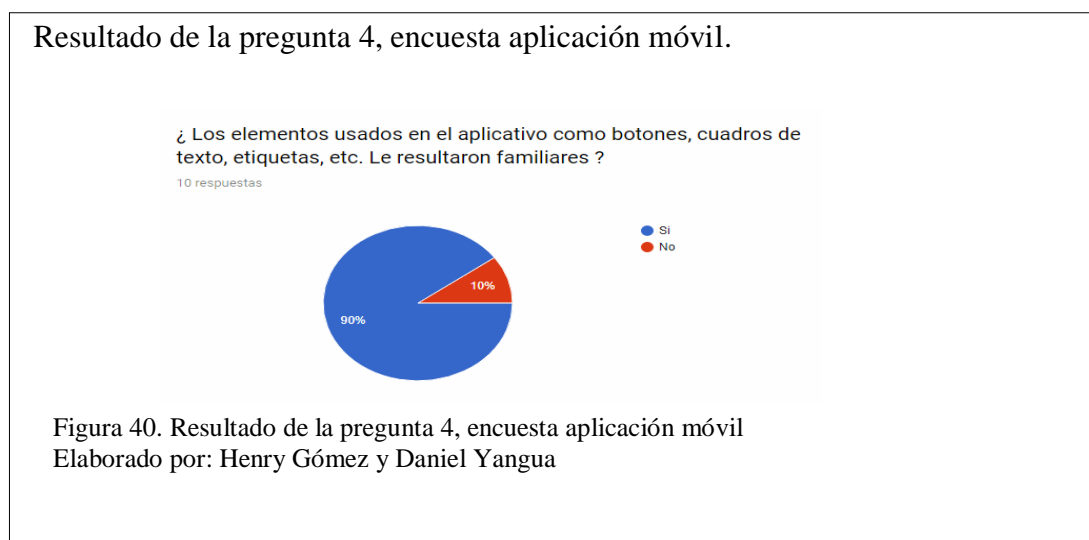
Como se muestra en la figura No 37 el diseño de aplicación móvil se trabajó bajo las guías de estilos recomendadas por Google Material, estas guías fueron creadas por expertos y tiene una gran aceptación dentro de la comunidad de desarrolladores Android. Permite usar buenas prácticas en cuanto a tamaños de fuente colores entre otras, se demuestra que la aplicación es aceptada con un 90% de los usuarios.



La experiencia de usuario es obtuvo un 80% como se aprecia en la figura No. 38, los procesos están pensados en ser simples y evitando interacciones innecesarias para el usuario.



En la figura No. 39 se demuestra el resultado que el 20% de usuarios percibían como un reto completar por primera vez la acción de iniciar un nuevo viaje, esto sucedió en el proceso de seleccionar las cooperativas y las unidades de transporte.



Solo el 10% de los usuarios no están familiarizados con los componentes usados como se puede apreciar en la figura No. 40. En la aplicación móvil se buscó brindar de una interfaz simple e intuitiva para los usuarios evitando distracciones en los colores y siguiendo un mismo estilo de apariencia con la aplicación web.

## CONCLUSIONES

- Se creó una estructura de datos en la aplicación móvil independiente a la de aplicación web, permitiendo que la información se enlace mediante servicios web que trabajen directamente con la base de datos.
- Se creó una tabla en la base de datos, en la cual se almacena la información procesada para mejorar el rendimiento en el tiempo de las consultas para la aplicación web.
- El uso de servicios en el aplicativo móvil permite ejecutar tareas en segundo plano, concediéndole al usuario el control para iniciar o detener el servicio, en el aplicativo se usa para iniciar detener un viaje en el momento que el usuario lo desee.
- Para almacenar la velocidad en la aplicación móvil, se creó un proceso en segundo plano, que permite iniciar o finalizar el servicio de la detección de velocidad.
- Un gran desafío del sistema fue diseñar su arquitectura, para ello se utilizó diagramas de componentes, el cual es un diagrama dentro del lenguaje de modelado UML, con ello se ha demostrado la necesidad y la utilidad de este lenguaje.
- Para llevar a cabo la consulta del promedio de la velocidad, se ha establecido el código dentro de la capa de modelo, en el patrón de arquitectura MVC, ya que la necesidad de consulta hacia la base de datos era de una gran concurrencia, de esta manera el sistema web se vuelve mucho más ágil, al momento de realizar este proceso.
- El uso de las pruebas permite identificar y mitigar los posibles fallos que se puedan presentar en las aplicaciones, tanto web, como móvil.

## **RECOMENDACIONES**

- Para los servicios web se recomienda el uso de un servidor NGIX configurado para el balance de carga para permitir una mejor respuesta y disponibilidad de los servicios en la salida a producción de los servicios.
- En una futura versión se puede añadir una aplicación en IOS usando los mismos servicios web para tener un mayor alcance de usuarios.
- Para establecer un control de velocidad más efectivo, se debe conocer los límites de velocidad por tramos en cada una de las carreteras, es por ello que se recomienda, generar dichos datos de todas las carreteras que se vaya a utilizar con la aplicación.
- Se recomienda el uso de procesos en segundo plano y se trate como servicio y no como una alarma de sistema que fuerza al dispositivo acceder al recurso, esto afecta directamente al consumo de batería en un dispositivo móvil.

## LISTA DE REFERENCIAS

- Agencia Nacional de Tránsito. (10 de 02 de 2014). *ant.gob.ec*. Obtenido de <https://www.ant.gob.ec/index.php/noticias/estadisticas>
- Alaix, V. G. (02 de 2007). *Ingeniería de Tránsito*. Obtenido de docentes.unal.edu.co: <http://www.docentes.unal.edu.co/vgvalenc/docs/GUIA%20Ingenieria%20de%20Transito%2001-07.pdf>
- Alaix, V. G. (Febrero de 2007). *Ingeniería de Tránsito*. Medellín: Universidad Nacional de Colombia Sede Medellín. Obtenido de <http://www.docentes.unal.edu.co/vgvalenc/docs/GUIA%20Ingenieria%20de%20Transito%2001-07.pdf>
- ANT. (16 de marzo de 2018). *ant.gob.ec*. Obtenido de <https://www.ant.gob.ec/index.php/descargable/file/5109-siniestros-febrero-2018>
- Apache.org. (26 de Septiembre de 2018). *jmeter.apache.org*. Obtenido de <https://jmeter.apache.org/>
- Basterra, Berte, Borello, Castillo, & Ventur. (2017). Android OS Documentation.
- BBVA. (23 de 03 de 2016). *bbvaopen4u.com*. Obtenido de <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>
- BBVA. (07 de 06 de 2016). *bbvaopen4u.com*. Obtenido de <https://bbvaopen4u.com/es/actualidad/que-es-una-api-y-que-puede-hacer-por-mi-negocio>
- BBVA. (06 de Junio de 2016). *bbvaopen4u.com*. Obtenido de <https://bbvaopen4u.com/es/actualidad/que-es-una-api-y-que-puede-hacer-por-mi-negocio>
- BEEVA. (6 de junio de 2016). *beevea.com*. Obtenido de <https://www.beevea.com/beevea-view/desarrollo/retrofit-una-libreria-para-desarrollo-android-y-java/>
- Booch, G., Rumbaugh, J., Jacobson, I., Martínez, J. S., & Molina, J. J. G. (1999). *El lenguaje unificado de modelado (Vol. 1)*. Madrid.
- Camps Paré, R., Casillas Santillán, L., Costal Costa, D., Ginestà, M., Escofet, C., & Mora, O. (2005). En *Base de datos* (pág. 176). Barcenola.
- comercio, E. (22 de agosto de 2018). *el comercio*. Obtenido de <https://www.elcomercio.com/actualidad/mtop-instalacion-nuevosradares-velocidad-ecuador.html>
- Developer Mozilla Org. (06 de 12 de 2017). *developer.mozilla.org*. Obtenido de [https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern\\_web\\_app\\_architecture/MVC\\_architecture](https://developer.mozilla.org/en-US/Apps/Fundamentals/Modern_web_app_architecture/MVC_architecture)

- EllisLab. (26 de Septiembre de 2018). *codeigniter.com*. Obtenido de <https://www.codeigniter.com/>
- Google. (26 de Septiembre de 2018). *developer.android.com*. Obtenido de <https://developer.android.com/guide/>
- Google. (26 de Septiembre de 2018). *developer.android.com*. Obtenido de <https://developer.android.com/studio/intro/?hl=es-419>
- Google. (25 de Abril de 2018). *developer.android.com*. Obtenido de <https://developer.android.com/training/permissions/requesting?hl=es-419>
- GU, M.-x., & TANG, K. (2010). Comparative analysis of WebForms MVC and MVP architecture. *2nd Conference on Environmental Science and Information Application Technology*, 392.
- IBM. (22 de 11 de 2014). *ibm.com*. Obtenido de <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Rational+Team+Concert+for+Scrum+Projects/page/SCRUM+como+metodolog%C3%ADa>
- IBM. (2016). *ibm.com*. Obtenido de <https://www.ibm.com/analytics/es/es/technology/data-science/>
- IBM. (26 de Septiembre de 2018). *ibm.com*. Obtenido de [https://www.ibm.com/support/knowledgecenter/es/SSZLC2\\_8.0.0/com.ibm.commerce.developer.doc/concepts/csdmvcdespat.htm](https://www.ibm.com/support/knowledgecenter/es/SSZLC2_8.0.0/com.ibm.commerce.developer.doc/concepts/csdmvcdespat.htm)
- IBM. (26 de Septiembre de 2018). *ibm.com*. Obtenido de <https://www.ibm.com/analytics/es/es/technology/data-science/>
- La Hora Ec. (19 de 5 de 2013). *lahora.com*. Recuperado el 9 de 2 de 2018, de <https://lahora.com.ec/noticia/1101523310/ecuador-segundo-pac3ads-en-muertes-por-accidentes>
- leaflet. (21 de agosto de 2018). *leafletjs.com*. Obtenido de <https://leafletjs.com/>
- Microsoft. (26 de Septiembre de 2018). *code.visualstudio.com*. Obtenido de <https://code.visualstudio.com/docs>
- Mozilla Foundation Org. (2016). *mozilla.org*. Obtenido de <https://mozillascience.github.io/open-data-primers/1.1-why-open-data.html>
- Mozilla.org. (2018 de Febrero de 18). *developer.mozilla.org*. Obtenido de <https://developer.mozilla.org/en-US/docs/Glossary/REST>
- Mozilla.org. (27 de Noviembre de 2017). *developer.mozilla.org*. Obtenido de <https://developer.mozilla.org/es/docs/Web/HTML>
- Mozilla.org. (22 de Julio de 2017). *developer.mozilla.org*. Obtenido de <https://developer.mozilla.org/es/docs/Learn/CSS>
- Mozilla.org. (8 de Junio de 2018). *developer.mozilla.org*. Obtenido de <https://developer.mozilla.org/en->



US/docs/Web/Apps/Fundamentals/Modern\_web\_app\_architecture/MVC\_architecture

Mozilla.org. (25 de Septiembre de 2018). *developer.mozilla.org*. Obtenido de <https://developer.mozilla.org/es/docs/Web/JavaScript>

Mozilla.org. (27 de Septiembre de 2018). *mozillascience.github.io*. Obtenido de <https://mozillascience.github.io/open-data-primers/1.1-why-open-data.html>

MySQL . (26 de Septiembre de 2018). *dev.mysql.com*. Obtenido de <https://dev.mysql.com/doc/refman/5.7/en/what-is-mysql.html>

nodejs. (09 de 02 de 2018). *nodejs.org*. Obtenido de <https://nodejs.org/es/>

Nodejs.org. (26 de Septiembre de 2018). *expressjs.com*. Obtenido de <https://expressjs.com/es/resources/glossary.html>

Oracle. (26 de Septiembre de 2018). *java.com*. Obtenido de [https://www.java.com/es/download/faq/whatis\\_java.xml](https://www.java.com/es/download/faq/whatis_java.xml)

ORG, G. (18 de 11 de 2016). *gnu.org*. Obtenido de <https://www.gnu.org/licenses/licenses.es.html>

Pressman, R. S., & Troya, J. M. (1988). *Ingeniería del software*.

Quetglás, G. M., Lobo, F. T., & Cerverón Lleó, V. (1995). Fundamentos de informática y programación. España: Valencia : V.J., D.L.

Quetglás, G. M., Lobo, F. T., & Cerverón Lleó, V. (1995). Fundamentos de informática y programación. España: Valencia : V.J., D.L.

Quetglás, G. M., Lobo, F. T., & Cerverón Lleó, V. (1995). Fundamentos de informática y programación. España: Valencia : V.J., D.L.

Quetglás, G. M., Lobo, F. T., & Cerverón Lleó, V. (1995). Fundamentos de Informática y Programación. España: Valencia : V.J., D.L.

Quetglás, G., Lobo, F., & Cerverón Lleó, V. (1995). Fundamentos de informática y programación. España: Valencia : V.J., D.L.

Reynoso, C. (2004). *Introducción a la Arquitectura de Software*. Buenos Aires.

secure.php.net. (26 de Septiembre de 2018). *secure.php.net*. Obtenido de <https://secure.php.net/manual/es/intro-what-is.php>

SPPAT. (21 de 06 de 2017). *Servicio Público para Pago de Accidentes de Tránsito*. Obtenido de [protecciontransito.gob.ec](http://www.protecciontransito.gob.ec/):  
<http://www.protecciontransito.gob.ec/servicios/consecuencias-del-exceso-de-velocidad/>

Telégrafo, E. (14 de 01 de 2015). *El Telégrafo*. Obtenido de <http://www.eltelegrafo.com.ec/noticias/judicial/13/100-radares-fijos-con-camaras-controlan-las-vias>

- Testea. (s.f.). *aragon.es/testea*. Obtenido de [https://www.aragon.es/estaticos/GobiernoAragon/Departamentos/IndustriaInnovacion/Areas/Sociedad\\_Informacion/Documentos/PROTESTEA-Curso%20Basico-Testing-03r003.pdf](https://www.aragon.es/estaticos/GobiernoAragon/Departamentos/IndustriaInnovacion/Areas/Sociedad_Informacion/Documentos/PROTESTEA-Curso%20Basico-Testing-03r003.pdf)
- Tinoco Gómez, O., Rosales López, P. P., & Salas Bacalla, J. (2010). Criterios de selección de metodologías de desarrollo de software. *Industrial Data*.
- Tránsito, A. N. (24 de 11 de 2016). *ANT*. Obtenido de <https://www.ant.gob.ec/index.php/transito-7/resoluciones-2016/file/3817-reglamento-de-homologacion-uso-y-validacion-de-sistemas-dispositivos-y-equipos-tecnologicos-para-la-detencion-y-notificacion-de-infracciones-de-transito?tmpl=component>
- Universidad de Carabobo. (2012). *Construcción y Pruebas de Software*. Burbula: Facultad Experimental de Ciencias y Tecnología Departamento de Computación .
- Universo, E. (19 de 11 de 2017). *El Universo*. Obtenido de <https://www.eluniverso.com/noticias/2017/11/19/nota/6486295/570906-multas-ir-velocidad-alta-4-anos>
- universo, E. (17 de Agosto de 2018). *eluniverso.com*. Obtenido de <https://www.eluniverso.com/noticias/2018/08/17/nota/6908768/tasa-mortalidad-accidentes-transito-ecuador-ha-aumentado-2018>

## ANEXOS

- Product Backlog – [Anexo 1](#)
- Sprint Backlog – [Anexo 2](#)
- Manual Técnico – [Anexo 3](#)
- Manual Usuario – [Anexo 4](#)
- Encuesta usabilidad aplicación móvil – [Anexo 5](#)
- Resultados encuesta usabilidad aplicación móvil – [Anexo 6](#)
- Encuesta usabilidad aplicación web - [Anexo 7](#)
- Resultados encuesta usabilidad aplicación web - [Anexo 8](#)